Claude DELANNOY

FAITES VOS JEUX

avec

AMSTRAD

EYROLLES

FAITES VOS JEUX AVEC AMSTRAD

FAITES VOS JEUX AVEC AMSTRAD

par

Claude DELANNOY



61, boulevard Saint-Germain — 75005 PARIS
1986

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service « Presse », Editions EYROLLES,
61, Boulevard Saint-Germain
75240 PARIS CEDEX 05,

en précisant les domaines qui vous intéressent. Vous recevrez régulièrement un avis de parution des nouveautés en vente chez votre libraire habituel.

Avant-propos

Ce livre vous propose un éventail de jeux exploitant pleinement les possibilités graphiques et sonores des AMSTRAD CPC 464, CPC 664 et CPC 6128.

Son rôle ne se limite cependant pas à celui d'un simple recueil de programmes. Tout d'abord, il vous offre la possibilité d'adapter chacun d'entre eux en fonction de vos goûts et de vos désirs, et cela sans même que vous n'ayez besoin de connaître le Basic. C'est dans ce but que chaque jeu est accompagné de nombreuses suggestions de personnalisation qui vous sont expliquées dans le moindre détail.

Par ailleurs, si vous avez quelques rudiments de Basic, ce livre peut constituer une bonne occasion de vous perfectionner en vous amusant et de vous initier aux méthodes particulières de programmation des jeux. C'est dans cette optique que nous avons préféré présenter les programmes par ordre de difficulté croissante plutôt que de les classer par "genre". Chacun d'entre eux est analysé instruction par instruction et les "techniques" employées sont clairement expliquées. Vous pourrez ainsi acquérir progressivement et agréablement les connaissances et le savoir-faire qui vous seront précieux pour développer vos propres jeux ou pour modifier profondément ceux qui vous sont proposés.

Les programmes ont été soigneusement vérifiés et leurs listes réalisées directement à partir de l'AMSTRAD. Vous pouvez donc les recopier en toute confiance.

Table des matières

Introduction

Ce livre peut s'utiliser de diverses manières. Afin que vous puissiez en tirer le meilleur profit, nous vous donnons ici quelques indications sur la façon dont il est constitué.

Un **premier chapitre** vous donne des conseils pour la frappe des programmes proposés. Si vous ne connaissez pas le Basic, sa lecture vous sera très profitable. Elle vous aidera à détecter les erreurs de frappe que vous aurez pu commettre lors de la recopie d'un des jeux.

Les **chapitres 2 à 19** correspondent chacun à un jeu. Ils sont tous structurés de la même manière en cinq paragraphes :

- *paragraphe 1 : le jeu.* Vous y trouverez le scénario du jeu, ses règles et tout ce qu'il faut savoir pour y jouer avec votre AMSTRAD. Généralement, vous y rencontrerez une photo noir et blanc (dans le texte) ou une référence à une photo couleur placée "hors-texte" ; dans les deux cas, elle présentera un exemple de l'écran pendant le déroulement d'une partie.

- *paragraphe 2 : le programme.* (Il s'agit de sa liste).

- *paragraphe 3 : si vous souhaitez personnaliser ce programme.* De nombreuses possibilités d'adaptation vous sont proposées. Elles ne requièrent absolument aucune connaissance du Basic. Dans le cas où cela présente de l'intérêt, nous vous indiquons comment utiliser une poignée de jeu.

- *paragraphe 4 : description du programme.* Vous y verrez tout d'abord la ''liste des techniques'' employées dans le programme. Il s'agit là de techniques spécifiques à la programmation des jeux. Chacune d'entre elles fait l'objet d'une description très détaillée en annexe. Cette façon de procéder nous a évité de répéter des explications nécessairement analogues d'un chapitre à un autre. Vient ensuite une analyse très complète du programme et des différents sous-programmes (dont nous avons fait un très large usage afin de mieux mettre en évidence le fonctionnement du jeu). Ces derniers sont décrits dans l'ordre où ils sont appelés dans le programme principal (et non pas nécessairement dans l'ordre des numéros de ligne). Notez que les structures de boucle sont mises en évidence par une typographie appropriée. D'autre part, nous avons, dans la mesure du possible, donné à chaque programme une structure comparable. Notamment, vous trouverez toujours un sous-programme d'initialisation du jeu (situé en 9000), un sous-programme d'initialisation de partie (situé en 8000)...

- *paragraphe 5 : liste des variables.* On y précise le rôle de chaque variable, à l'exception de quelques unes, rangées dans la rubrique ''variables auxiliaires'' ; ce sont là des variables n'ayant qu'un rôle secondaire et qui n'apparaissent que très localement : compteur de boucle FOR ayant peu d'instructions, variable chaîne destinée à lire une réponse O/N, etc...

Viennent enfin les différentes **annexes** dont nous avons parlé plus haut. Leur ensemble constitue une initiation à la programmation des jeux. Vous y apprendrez comment jouer avec le hasard, animer des mobiles, les commander par le clavier et vous verrez comment connaître le contenu d'un emplacement quelconque de l'écran.

1

Conseils pour la frappe des programmes

Ce petit chapitre vous donne quelques conseils pour faciliter cette opération souvent laborieuse qu'est la frappe des programmes.

1. Sachez profiter de l'AUTO

N'hésitez pas (si vous la connaissez) à utiliser la commande AUTO qui vous évite d'avoir à taper les numéros de ligne. Nos listes sont numérotées de 10 en 10 (sauf erreur de notre part), avec changement de numérotation à chaque sous-programme.

Il vous suffit donc de taper :

```
AUTO 100
```

avant de commencer la frappe. Vous obtiendrez ainsi "automatiquement" les numéros de ligne : 100, 110, 120, etc...

A la fin de l'entrée de chaque sous-programme (c'est-à-dire quand apparaît dans la liste, une ligne comportant beaucoup de tirets), vous quitterez le mode AUTO en frappant deux fois la touche ESC. Puis, vous exécuterez à nouveau la commande AUTO avec un nouveau numéro ; par exemple :

AUTO 8000

vous fournira automatiquement les numéros 8000, 8010, 8020, ...

2. *Pensez aux "sauvegardes"*

Bien entendu, avant de commencer à exécuter votre programme, il est nécessaire de le vérifier attentivement à l'écran ou sur imprimante (si vous avez la chance d'en posséder une). Mais il est en outre indispensable que vous en fassiez une sauvegarde sur cassette ou sur disquette avant de commencer à l'exécuter.

En effet, si pour une raison quelconque, votre AMSTRAD se "plante", vous n'avez pas d'autre ressource que de "revenir à l'état initial" à l'aide des touches CTRL, SHIFT et ESC (on garde CTRL et SHIFT enfoncées, tandis qu'on appuie sur ESC). Cela produit le même résultat qu'une coupure momentanée de votre AMSTRAD. Dans ces conditions, vous voyez que vous perdez ainsi le programme que vous avez patiemment entré en mémoire. Si vous avez pris soin de le sauvegarder au préalable, il vous suffit de le recharger puis d'en rechercher les anomalies.

Notez que cette technique est également utilisable pour des sauvegardes "partielles". Celles-ci sont conseillées lorsque :

- vous n'avez pas le temps de taper tout votre programme en une seule fois,
- vous craignez d'être interrompu par une "fausse manip", une coupure secteur,...
- vous souhaitez faire une "mise au point progressive" comme indiqué ci-après.

3. *Recherche des anomalies*

Il se peut que, malgré vos vérifications scrupuleuses, votre programme ne fonctionne pas au premier coup. Certes, dans ce cas, vous pouvez continuer à effectuer une vérification ligne à ligne de l'ensemble du programme. Cependant, la plupart du temps, il vous sera possible de mieux cerner l'anomalie, en fonction des "messages" fournis par Basic.

a) Lorsque vous obtenez un message du type :

```
Syntax error in xxxx
```

(où xxxx désigne un numéro de ligne)

Celui-ci signifie ''erreur de syntaxe en xxxx''. Ici, aucun doute n'est permis. La ligne en question comporte une erreur. Une simple comparaison avec la liste doit vous permettre de la déceler.

Notez qu'à la suite de ce message, Basic vous affiche la ligne coupable et vous met automatiquement en mode ''édition'' sur cette ligne (cela se reconnaît à la présence du curseur.

Si vous savez comment effectuer des corrections dans ce mode (touches ESC, CLR, ...), vous pouvez le faire directement. Si ce n'est pas le cas, sachez qu'il vous suffit alors de taper ''ENTER'' pour quitter ce mode et de taper à nouveau l'instruction en question (précédée de son numéro de ligne).

b) Lorsque vous obtenez un message du type

```
Improper argument in xxxx
```

Cela signifie qu'en ligne xxxx, vous employez une instruction ou une fonction à laquelle vous fournissez des valeurs qui ne conviennent pas (par exemple un numéro supérieur à 26 pour PAPER, une valeur négative pour LOCATE, etc...).

Commencez bien entendu par vérifier la ligne concernée. Si elle ne comporte pas d'erreur, vous pouvez en déduire que certaines ''variables'' (apparaissant dans cette instruction) ont pris de mauvaises valeurs.

Si vous connaissez un peu le Basic et la fonction concernée, vous pouvez faire écrire, en *mode direct*, le contenu des variables. Par exemple, si l'instruction est :

```
LOCATE X,Y
```

vous pouvez, après le message d'erreur (et après avoir pressé ''ENTER'' pour quitter le mode édition) taper simplement, *sans numéro de ligne* :

```
PRINT X,Y
```

Vous devriez ainsi découvrir la variable coupable. Il vous faut ensuite vérifier toutes les instructions dans lesquelles la variable (ou les variables si, ne connaissant pas Basic, vous ne savez pas laquelle est en cause) reçoit une valeur. Ici, par exemple, vous vérifierez toutes les instructions de la forme :

```
    X = ...
ou  Y = ...
```

D'autres erreurs sont malheureusement possibles. Celles que nous avons évoquées sont les plus courantes. Si vous prenez bien soin de relire ce que vous avez tapé et de le comparer avec la liste d'origine, il est probable que vous ne rencontrerez guère d'autres erreurs.

4. La mise au point progressive

Si vous connaissez le Basic, vous pouvez profiter de la structure ''très modulaire'' des programmes que nous vous proposons.

Par exemple, vous pouvez entrer d'abord le programme principal (généralement court) puis les différents sous-programmes en tenant compte de l'ordre dans lequel ils sont utilisés. Généralement, vous commencerez par le sous-programme d'initialisation du jeu (situé en 9000). Celui-ci peut d'ailleurs être testé par un programme principal réduit à

```
GOSUB 9000
```

Vous poursuivrez votre ''saisie' par le sous-programme suivant (généralement : initialisation d'une partie en 8000) et ainsi de suite. L'ordre dans lequel il faut entrer les différents sous-programmes est celui dans lequel ils sont mentionnés dans la rubrique ''description du programme''.

2

Force de frappe

1. Le jeu

Voici un bon moyen d'apprendre à mieux connaître le clavier de votre AMSTRAD et d'augmenter ainsi votre vitesse de frappe. En effet, le but de ce jeu consiste à taper le plus rapidement possible sur la touche correspondant à la lettre que vous voyez apparaître en un endroit quelconque de l'écran.

A chaque partie, trente lettres vous sont ainsi proposées. Chacune d'entre elles est tirée au hasard parmi les vingt-six lettres de l'alphabet. Si vous frappez la bonne touche, vous serez ''récompensés'' par un son qui deviendra de plus en plus ''haut'' au fur et à mesure de votre progression. Par contre, si vous vous trompez de touche, un son approprié vous le signalera ; il en ira ainsi jusqu'à ce que vous tapiez sur la bonne touche.

Les lettres sont proposées suivant un rythme irrégulier. C'est là un petit détail qui accroît quelque peu la difficulté du jeu en vous obligeant d'être sans cesse ''à l'affût'' d'une nouvelle lettre.

Vous verrez s'afficher en permanence le nombre d'erreurs et le temps écoulé depuis le début de la partie. Seul n'est pris en compte que votre temps de réaction. Le temps (variable) qui s'écoule avant l'apparition de chaque lettre n'est pas considéré.

A la fin de chaque partie, le programme vous indique votre temps moyen de réaction. Il vous délivre alors un score qui dépend à la fois de ce temps moyen et du nombre d'erreurs commises. En même temps, il vous renseigne sur le meilleur score réalisé depuis le début du jeu.

Pour rejouer, il vous suffit de répondre O (pour oui) à la question : ''voulez-vous rejouer (O/N) ?''

Remarque importante : assurez-vous que votre clavier est en ''mode'' majuscule.

2. Le programme

```
100 '******************* FORCE DE FRAPPE **********************
110 '
120 GOSUB 9000
130  GOSUB 8000
140  FOR L = 1 TO NL
150   GOSUB 5000
160   R$=INKEY$: IF R$="" THEN GOSUB 6000: GOTO 160
170   R=ASC(R$): IF R<>C THEN GOSUB 3000: GOTO 160
180   GOSUB 2000
190  NEXT L
200  M=T/NL
210  LOCATE 1,8: PRINT "TEMPS MOYEN"; M
220  LOCATE 1,11: PRINT "AVEC"; E; "ERREURS";
230  SC=INT(30000/M-50*E)
240  LOCATE 1,16: PRINT "VOTRE SCORE"; SC
250  LOCATE 1,21: PRINT "MEILLEUR SCORE"; MS
260  IF SC>MS THEN MS=SC
270  LOCATE 1,24
280  PRINT "voulez vous rejouer (O/N)"
290  R$=INKEY$: IF R$<>"N" AND R$<>"O" THEN GOTO 290
300 IF R$="O" THEN GOTO 130
310 END
```

```
320 '
2000 '------------ SP LETTRE CORRECTE -----------------------
2010 '
2020 LOCATE X,Y: PRINT " ";
2030 KN=KN+1
2040 SOUND 1,150-3*KN,10
2050 RETURN
2060 '
3000 '------------ SP LETTRE INCORRECTE ---------------------
3010 '
3020 E=E+1: SOUND 1,2000,30
3030 LOCATE 22,2: PRINT E;
3040 KN=0
3050 RETURN
3060 '
5000 '---------- SP AFFICHAGE LETTRE ------------------------
5010 '
5020 X=X1+INT(RND(1)*(X2-X1+1))
5030 Y=Y1+INT(RND(1)*(Y2-Y1+1))
5040 IF X=40 AND Y=25 THEN 5020
5050 C=65+INT(RND(1)*26)
5060 FOR K=1 TO RND(1)*TM: NEXT K
5070 LOCATE X,Y: PRINT CHR$(C);
5080 RETURN
5090 '
6000 '------------ SP TEMPS ---------------------------------
6010 '
6020 T=T+1
6030 LOCATE 9,2: PRINT T;
6040 RETURN
6050 '
8000 '------------ SP INIT PARTIE ---------------------------
8010 '
8020 CLS: T=0: E=0
8030 PRINT "pour commencer, tapez sur une touche";
8040 R$=INKEY$
8050 IF R$="" THEN GOTO 8040
8060 CLS
8070 LOCATE 3,2: PRINT "TEMPS       ERREURS";
8080 KN=0
8090 RETURN
8100 '
```

```
9030 X1=1: X2=40: Y1=4: Y2=25
9040 NL=30: TM=500: MS=0
9050 INK 0,CF: 'INK 1,CE: BORDER BR
9060 RETURN
9000 '------------ SP INIT JEU --------------------------------
9010 '
9020 CF=23: CE= 2: BR=16
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier les couleurs :**

— pour modifier la couleur de fond, remplacez, en 9020, le nombre 23 par une valeur de votre choix (de 0 à 26),

— pour modifier la couleur d'écriture, remplacez, en 9020, le nombre 2 par une valeur de votre choix (de 0 à 26). Évitez de choisir les mêmes couleurs pour le fond et l'écriture ; vos lettres seraient alors "invisibles",

— pour modifier la couleur de la bordure de l'écran, remplacez, en 9020, le nombre 16 par une valeur de votre choix (de 0 à 26).

▷ **Pour modifier le nombre de lettres proposées lors d'une partie**

Remplacez, en 9040, le nombre 30 par la valeur de votre choix.

▷ **Pour modifier le temps s'écoulant avant l'apparition d'une nouvelle lettre**

Remplacez, en 9040, le nombre 500 par une valeur de votre choix. Si vous souhaitez supprimer le temps d'attente, vous pouvez :

— soit remplacer, en 9040, la valeur 500 par 0,

— soit supprimer l'instruction 5060.

▷ **Pour que les lettres s'affichent toujours au même endroit**

Remplacez les lignes 5020 et 5030 par des instructions attribuant des valeurs déterminées à X et à Y. Par exemple :

```
5020 X = 20
5030 Y = 12
```

placeront toutes les lettres au centre de l'écran.

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard.

b) Le programme principal

120 appelle le sous-programme d'initialisation du jeu (9000).

130-300 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous disiez que vous ne souhaitez plus rejouer :

140-190 proposent les trente lettres. Pour chaque lettre :

150 appelle le sous-programme 5000 qui affiche une lettre à l'écran.

160 attend qu'une touche soit pressée. Tant que cela n'a pas lieu, le sous-programme 6000 est appelé pour ''incrémenter'' le compteur de temps et pour en afficher la valeur.

170 analyse la réponse. Si celle-ci est incorrecte, le sous-programme ''lettre incorrecte'' (3000) est appelé ; on revient alors en 160 pour attendre une autre proposition.

180 exécutée lorsque la réponse est exacte ; cette instruction appelle le sous-programme ''lettre correcte''.

200-220 calculent le temps moyen et l'affichent en même temps que le nombre d'erreurs.

230 calcule le score.

240-250 affichent votre score et le meilleur score déjà réalisé.

260 actualise le meilleur score, s'il y a lieu.

270-290 vous demandent si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9060)

9020 détermine les couleurs (fond, écriture, bordure).

9030 fixe les limites du domaine où peuvent apparaître les lettres proposées.

9040 fixe le nombre de lettres proposées, le temps d'attente maximum et initialise le meilleur score.

d) Le sous-programme d'initialisation d'une partie (8000-8090)

8020 efface l'écran et initialise les compteurs de temps et d'erreurs.

8030-8060 attendent que vous frappiez une touche et effacent l'écran.

8070 affiche les textes accompagnant le temps, le nombre de lettres affichées et le nombre d'erreurs.

8080 initialise le nombre de réponses consécutives sans erreur.

e) le sous-programme d'affichage d'une lettre (5000-5080)

5020-5040 déterminent au hasard les coordonnées d'affichage d'une lettre (voir annexe hasard). Le point situé en bas à droite de l'écran (X = 40, Y = 25) est rejeté car l'écriture d'une lettre à cet emplacement provoquerait un passage à la ligne avec défilement d'écran ; la ligne supérieure disparaîtrait alors.

5050 choisit au hasard le code ASCII de l'une des vingt six lettres (majuscules) de l'alphabet (voir annexe hasard).

5060-5070 affichent la lettre choisie à l'emplacement déterminé après un temps tiré lui aussi au hasard.

f) Le sous-programme lettre correcte (2000-2050)

2020 efface la lettre proposée.

2030 incrémente le nombre de réponses consécutives sans erreur.

2040 émet un son dont la hauteur dépend du nombre de réponses consécutives sans erreur.

g) Le sous-programme lettre incorrecte (3000-3050)

3020 incrémente de un le compteur d'erreurs et émet un son approprié.

3030 affiche le nombre d'erreurs.

3030 remet à 1 le compteur de réponses consécutives sans erreur.

5. Liste des variables

CF	Couleur de fond.
CE	Couleur d'écriture.
BR	Couleur de la bordure de l'écran.
X1,X2,Y1,Y2	Coordonnées du domaine dans lequel les lettres pourront apparaître.
NL	Nombre de lettres proposées au cours d'une partie.

TM	Temps maximum s'écoulant avant l'apparition d'une nouvelle lettre.
MS	Meilleur score.
T	Compteur du temps (cumulé) de réaction du joueur.
E	Compteur d'erreurs.
KN	Nombre de réponses *consécutives* sans erreur (il est remis à zéro à chaque erreur).
L	Numéro de la lettre courante.
X,Y	Coordonnées de la lettre courante.
C	Code ASCII de la lettre courante.
R	Code ASCII de la touche frappée par le joueur.
M	Temps moyen de réaction.
SC	Score.

Variables auxiliaires

R$ K

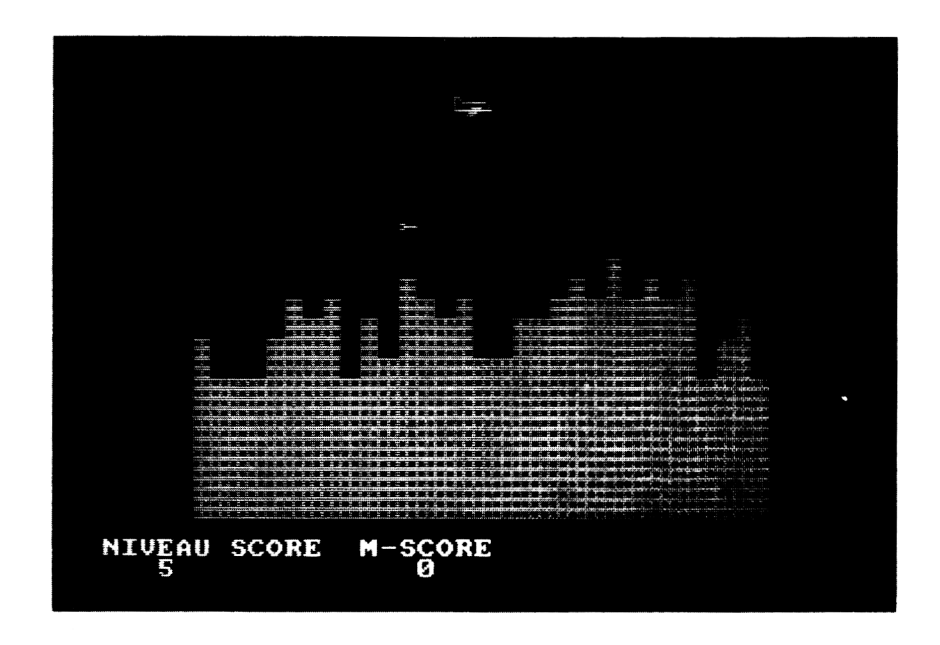
3
Raid de nuit

1. Le jeu

En pleine nuit, vous êtes aux commandes d'un bombardier. Votre mission : anéantir une ville. Le temps vous est compté car votre moteur est défaillant et vous perdez sans cesse de l'altitude. Vous avez donc intérêt à démolir rapidement les immeubles les plus élevés pour éviter de les percuter lors d'un prochain passage. D'autre part, vous ne pouvez larguer une nouvelle bombe que lorsque la précédente a atteint son but (ou le sol).

Vous voyez donc votre avion traverser l'écran à diverses reprises. A chaque passage, il peut perdre ou ne pas perdre (suivant le hasard) un peu d'altitude. Le ''suspens'' en sera ainsi d'autant plus grand quand vous serez passés au ras d'un immeuble que vous aurez raté !

Vous larguez une bombe en pressant n'importe quelle touche. Chaque fois qu'un



immeuble est atteint, il est démoli sur une hauteur plus ou moins importante. Des sons appropriés vous renseignent sur l'importance des dégâts.

Le jeu s'achève lorsque votre avion percute un immeuble ou lorsque la ville a été entièrement détruite. Dans le premier cas, votre score est fonction de l'importance des destructions opérées. Dans le second cas, il dépend de la vitesse avec laquelle vous avez atteint votre objectif (il sera toujours plus élevé dans le second cas).

Neuf niveaux différents vous sont proposés. Au niveau 1, les immeubles sont de petite taille tandis qu'au niveau 9, ils occupent une bonne partie de l'écran.

Le programme affiche en permanence le niveau auquel vous jouez et le meilleur score déjà réalisé à ce niveau.

2. Le programme

```
100 '**************** RAID DE NUIT ***************************
110 '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+1,16*(25-Y)+1)
130 GOSUB 9000
140  GOSUB 8000
150   IF TR=1 THEN GOSUB 2000
160   R$=INKEY$
170   IF R$<>"" AND TR=0 THEN XB=XA: YB=YA: TR=1
180   GOSUB 3000
190  IF TR=1 AND EX=0 THEN GOSUB 2000 ELSE FOR K=1 TO 20:NEXT K
200   IF FI<>1 AND ID<>IM THEN 150
210  IF ID=IM THEN GOSUB 5000
220  LOCATE #1, 10,3: PRINT #1,SC;
230  IF SC>MS(NV) THEN MS(NV)=SC
240  FOR I=1 TO 1000: NEXT I
250  LOCATE #1, 24,1: PRINT #1, "voulez vous";
260  LOCATE #1, 24,2: PRINT #1, "rejouer (O/N)";
270  R$=INKEY$: IF R$<>"N" AND R$<>"O" THEN 270
280 IF R$="O" THEN 140
290 END
300 '
2000 '------------ SP DEPLACEMENT BOMBE ----------------------
2010 '
2020 IF EX=1 THEN 2070
2030 LOCATE XB,YB: PRINT " ";: YB=YB+1
2040 IF YB>Y2 THEN SOUND 1,2000,10: TR=0: RETURN
2050 IF FNSC(XB,YB)=CF THEN LOCATE XB,YB: PRINT GB$;: RETURN
2060 EX=1
2070 SOUND 1,200,5: LOCATE XB,YB: PRINT " ";
2080 ID=ID+1: YB=YB+1
2090 IF RND<ZP OR YB>Y2 THEN EX=0: TR=0
2100 RETURN
2110 '
3000 '------------ SP DEPLACEMENT AVION ----------------------
3010 '
3020 IF XA>=X2 THEN 3050
3030 IF FNSC(XA+3,YA)<>CF THEN GOSUB 4000: RETURN
3040 LOCATE XA,YA: PRINT GA$;: XA=XA+1: RETURN
3050 LOCATE XA,YA: PRINT "  ";: IF RND>0.33 THEN YA=YA+1
3060 IF YA>Y2 THEN GOSUB 5000
3070 XA=X1
```

```
3080 LOCATE XA,YA: PRINT A$;
3090 RETURN
3100 '
4000 '------------ SP FIN DE PARTIE PAR AVION DETRUIT --------
4010 '
4020 LOCATE XA,YA: PRINT "  ";
4030 FOR I=1 TO 15
4040  LOCATE XA,YA: PEN I: PRINT GA$;
4050  SOUND 1,10*I,5
4060 NEXT I
4070 FI=1
4080 LOCATE XA+1,YA: PRINT "  ";
4090 SC=INT(1000*ID/IM)
4100 RETURN
4110 '
5000 '------------ SP FIN DE PARTIE PAR VILLE DETRUITE -------
5010 '
5020 SC=1000+(Y2-YA)*100
5030 RETURN
5040 '
8000 '------------ SP INITIALISATION D'UNE PARTIE ------------
8010 '
8020 PAPER 0: CLS
8030 PEN #1,3: CLS #1
8040 LOCATE #1, 3,1
8050 PRINT #1, "niveau choisi (1 a 9)";
8060 R$=INKEY$: NV=VAL(R$)
8070 IF NV=0 THEN 8060
8080 PRINT #1, NV;
8090 '
8100 PEN 1
8110 YB=Y2-NV-1: HM=0.5*NV+4: IM=0
8120 FOR X=X1+3 TO X2-1
8130  YM=YB-INT(RND(1)*HM)
8140  IM=IM+Y2-YM+1
8150  FOR Y=Y2 TO YM STEP -1
8160   LOCATE X,Y: PRINT CHR$(163);
8170  NEXT Y
8180 NEXT X
8190 '
8200 ID=0: TR=0: EX=0: FI=0
8210 XA=X1: YA=Y1: XB=XA: YB=YA
8220 PEN 2: LOCATE XA,YA: PRINT GA$;
8230 CLS #1
```

```
8240 LOCATE #1, 2,2: PRINT #1,"NIVEAU SCORE  M-SCORE";
8250 LOCATE #1, 4,3: PRINT #1, NV;
8260 LOCATE #1, 18,3: PRINT #1, MS(NV);
8270 RETURN
8280 '
9000 ' ------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 BR=0: CF=0: CI=4 :CA=6: CS=16
9030 INK 0,CF: INK 1,CI: INK 2,CA: INK 3,CS
9040 MODE 1: BORDER BR
9050 X1=4: X2=38: Y1=1: Y2=22
9060 WINDOW #1, 1,40,Y2+1,25
9070 ZP=0.6
9080 '
9090 SYMBOL AFTER 128
9100 SYMBOL 160,224,160,159,128,129,255,3,7
9110 SYMBOL 161,0,0,252,2,242,255,192,128
9120 SYMBOL 162,0,0,224,63,224,0,0,0
9130 SYMBOL 163,255,153,153,255,255,153,153,255
9140 A$=CHR$(160)+CHR$(161)
9150 GA$=" "+A$
9160 GB$=CHR$(162)
9170 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

▷ **Pour modifier les couleurs**

— *de la bordure* (ici noire) : remplacez, en 9020, le nombre 0 (dans BR = 0) par une valeur de votre choix (entre 0 et 26),

— *du ciel* (ici noir) : remplacez, en 9020, le nombre 0 (dans CF = 0) par une valeur de votre choix,

— *des immeubles :* remplacez, en 9020, le nombre 4 par une valeur de votre choix,

— *de l'avion et de la bombe :* remplacez, en 9020, le nombre 6 par une valeur de votre choix.

▷ **Pour que l'avion perde de l'altitude plus ou moins rapidement**

La perte d'altitude est régie par la ligne 3050. Actuellement, à chaque passage, l'avion a une chance sur trois (0.33) de ne pas descendre.

Si vous souhaitez que l'avion descende plus vite, remplacez cette valeur 0.33 par un nombre plus petit. Par exemple, avec :

```
3050 LOCATE XA,YA : PRINT "  " ; IF RND > 0.1 THEN YA = YA + 1
```

l'avion n'aura plus qu'une chance sur dix (0.1) de ne pas descendre. Il descendra donc, en moyenne neuf fois sur dix.

Si vous voulez que l'avion descende systématiquement à chaque passage, il vous suffit d'écrire :

```
3050 LOCATE XA,YA : PRINT "  " : YA = YA + 1
```

Enfin, si vous désirez que l'avion descende plus lentement, remplacez la valeur 0.33 par un nombre plus grand (*mais inférieur à 1*). Par exemple, avec :

```
3050 LOCATE XA,YA : PRINT "  " : IF RND > 0.5 THEN YA = YA + 1
```

l'avion descendra, en moyenne, une fois sur deux.

Évitez de choisir des valeurs trop proches de 1 car alors l'avion descendrait trop lentement et le jeu risquerait de perdre de l'intérêt.

▷ **Pour modifier le rythme de déplacement de l'avion**

Actuellement, à chaque largage de bombe, l'avion ralentit comme pour laisser au pilote la possibilité d'examiner les résultats obtenus. Dès que la bombe a atteint son objectif, l'avion reprend sa vitesse normale.

Vous pouvez obtenir un déplacement à ''vitesse constante'' en modifiant la durée de l'attente en ligne 190 ; pour cela, il vous suffit de remplacer la valeur 20 par 50.

4. *Description du programme*

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Examen du contenu de l'écran.

b) Le programme principal (100-290)

120 définit une fonction permettant de connaître la couleur (numéro d'encrier) d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-280 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus réjouer :

140 appelle le sous-programme d'initialisation d'une partie.

150-200 sont répétées jusqu'à la fin de la partie (écrasement de l'avion ou destruction de toute la ville) :

150 assure, à l'aide du sous-programme 2000, le déplacement de la bombe si un largage est en cours (TR = 1).

160-170 examinent le clavier et prennent en compte une demande de largage (touche quelconque pressée) si aucun autre largage n'est en cours (TR = 0).

180 appelle le sous-programme de déplacement de l'avion.

190 appelle, s'il y a lieu (tir en cours et explosion d'immeuble non en cours) le sous-programme de déplacement de la bombe.

200 examine si la partie est achevée.

210 appelle, s'il y a lieu, le sous-programme de fin de partie par destruction totale de la ville (notez que le sous-programme de fin de partie par écrasement de l'avion est, quant-à-lui, appelé par le sous-programme de déplacement de l'avion).

220-230 affichent votre score et actualisent le meilleur score.

240-270 vous demandent si vous voulez rejouer.

c) Le sous-programme d'initialisaton du jeu (9000-9180)

9020-9040 fixent les différentes couleurs

9050 fixe les limites du domaine ou évolue l'avion

9060 définit la fenêtre numéro 1 utilisée pour les différents affichages en bas de l'écran.

9070 fixe (dans ZP) la probabilité qu'une explosion d'une partie d'immeuble entraîne l'explosion de la partie inférieure.

9090-9160 fabriquent les caractères graphiques utilisés pour représenter l'avion, la bombe et les immeubles.

d) Le sous-programme d'initialisation de partie (8000-8270)

8020 efface l'écran.

8030-8080 vous font choisir votre niveau de jeu.

8100-8180 dessinent les immeubles (en comptabilisant dans IM le nombre d'emplacements correspondant).

8200 initialise le nombre d'immeubles détruits et les indicateurs de largage, d'explosion d'immeuble et de fin de partie.

8210 initialise les coordonnées de l'avion et de la bombe.

8220 dessine l'avion dans sa position initiale.

8230-8260 affichent le niveau de jeu et le meilleur score.

e) Le sous-programme de déplacement de la bombe (2000-2100)

(Ce sous-programme gère également l'explosion d'immeuble)

2020 examine si une explosion d'immeuble est en cours.

2030-2060 avancent la bombe (lorsqu'aucune explosion d'immeuble n'est en cours) et examinent si elle atteint le sol ou un immeuble. Dans ce dernier cas, l'indicateur EX (explosion en cours) est mis en place.

2070-2090 exécutées lorsqu'une explosion est en cours, ces instructions incrémentent le compteur d'immeubles détruits, effacent un ''morceau'' d'immeuble et déterminent si l'explosion doit ou non se poursuivre sur le partie inférieure.

f) Le sous-programme de déplacement de l'avion

3020-3040 déplacent l'avion s'il n'a pas atteint le bord de l'écran en examinant s'il percute un immeuble. Dans ce dernier cas, le sous-programme ''fin de partie par avion détruit'' (4000) est appelé.

3050-3090 exécutées lorsque l'avion a atteint le bord droit, ces instructions réaffichent l'avion à gauche de l'écran après avoir modifié, éventuellement, son altitude ; si l'avion atteint alors le sol le sous-programme ''fin de partie par ville détruite'' (5000) est appelé.

g) Le sous-programme de fin de partie-ville détruite (5000-5030)

Il se contente de calculer votre score.

h) Le sous-programme de fin de partie-avion détruit (4000-4100)

4020-4060 correspondent à la destruction de l'avion avec effets de sons et de couleurs.

4070-4080 positionnent l'indicateur de fin de partie et effacent l'avion.

4090 calcule le score.

5. *Liste des variables*

BR	Couleur du tour de l'écran.
CF	Couleur du fond (ciel et partie affichage des scores).
CI	Couleur des immeubles.
CA	Couleur de l'avion et de la bombe.
CS	Couleur d'affichage des scores et du ''dialogue''.
X1,Y1,X2,Y2	Coordonnées du domaine dans lequel évolue l'avion. (Notez que les immeubles sont tracés entre X1 + 3 et X2 − 1).
ZP	Probabilité pour que l'explosion d'une partie d'immeuble entraîne l'explosion de la partie inférieure.
A$	Chaîne de 2 caractères représentant l'avion.
GA$	Chaîne formée d'un espace et de deux caractères ''avion''. Elle permet de réaliser, à l'aide d'une seule instruction PRINT, le déplacement de l'avion.
GB$	Chaîne d'un caractère représentant la bombe.
NV	Niveau de jeu (de 1 à 9).
IM	Nombre total de ''morceaux d'immeubles''.
ID	Nombre de ''morceaux d'immeubles'' (cases) détruits.
TR	Indicateur de tir : 0 : aucun largage en cours, 1 : largage en cours (une bombe est en train de tomber ou un immeuble est en train d'exploser).
EX	Indicateur d'explosion d'immeuble : 0 : pas d'explosion, 1 : explosion en cours (un morceau d'immeuble vient d'être détruit et le ''hasard'' a décidé que le suivant devra l'être également).
FI	Indicateur de fin de partie : 0 : cas normal, 1 : la partie ne peut plus se poursuivre (ville détruite ou avion détruit).
XA,YA	Coordonnées de l'avion.
XB,YB	Coordonnées de la bombe.

MS(9)	Tableau contenant le meilleur score relatif à chacun des neuf niveaux.
SC	Score obtenu lors d'une partie.

Variables auxiliaires

R$ A X Y

YB,HM (dans le sous-programme 8000 seulement).

4

FLIP FLOP

1. Le jeu

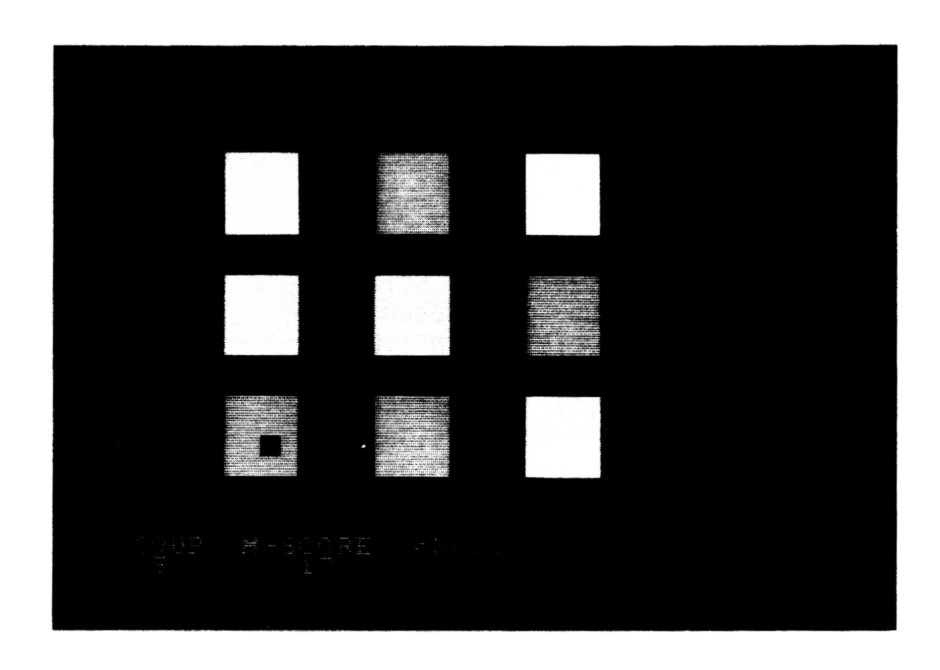
Vous disposez de neuf pions réversibles, disposés suivant une grille 3 × 3. L'une de leur face est bleue tandis que l'autre est verte.

Au départ, le programme dispose les pions dans la grille, en tirant au hasard la couleur de la face visible. Le but du jeu consiste à aboutir, en un minimum de coups à la situation suivante :

— le pion central est vert,

— les autres pions sont bleus.

A chaque coup, vous choisissez le pion sur lequel vous voulez agir. Le programme vous retourne alors, non seulement ce pion, mais également un certain nombre d'autres, suivant les règles suivantes :



Pion choisi	*Pions retournés en plus du pion choisi*
Le pion central.	Les quatre pions des coins.
Un pion situé en milieu d'une rangée horizontale ou verticale.	Les deux pions voisins situés sur la même rangée.
Un pion situé dans un coin.	Les trois voisins immédiats.

Malgré une lointaine ressemblance avec le ''rubik's cube'', vous verrez que ce jeu est beaucoup plus facile à pratiquer.

Le choix d'un pion se fait à l'aide d'un ''curseur'' représenté par un carré noir. Il se déplace de pion en pion à l'aide des quatre touches ↓↑ → ←.

Le programme vous affiche en permanence le numéro du coup que vous jouez et le meilleur score déjà obtenu.

2. Le programme

```
100 '******************** F L I P - F L O P *****************
110 '
120 GOSUB 9000
130  GOSUB 8000
140   NC=NC+1
150   PEN 1: LOCATE 4,24: PRINT NC;
160   GOSUB 7000: GOSUB 6000: GOSUB 5000
170   IF NF<>8 OR T(2,2)<>1 THEN 140
180  IF MC<MS OR MS=0 THEN MS=NC
190  PEN 1: LOCATE 25,22: PRINT "GAGNE";
200  SOUND 1,478,30: SOUND 1,379,30
210  SOUND 1,319,30: SOUND 1,239,50
220  LOCATE 21,23: PRINT "rejouez vous?";
230  WHILE INKEY$<>"": WEND
240  R$=INKEY$: IF R$="" THEN 240
250 IF R$<>"N" THEN 130
260 END
270 '
5000 '------------ SP ANALYSE DU COUP -----------------------
5010 '
5020 NF=0
5030 FOR I=1 TO 3: FOR J=1 TO 3
5040  IF T(I,J)=-1 THEN NF=NF+1
5050 NEXT J: NEXT I
5060 RETURN
5070 '
6000 '------------ SP JEU DU COUP ---------------------------
6010 '
6020 IP=I: JP=J
6030 IF IP<>2 OR JP<>2 THEN 6110
6040  GOSUB 6500
6050  FOR I=1 TO 3 STEP 2
6060   FOR J=1 TO 3 STEP 2
6070    GOSUB 6500
6080   NEXT J
6090  NEXT I
6100 '
6110 IF IP<>2 THEN 6150
6120  FOR I=1 TO 3: GOSUB 6500: NEXT I
6130  RETURN
6140 '
```

```
6150 IF JP<>2 THEN 6190
6160  FOR J=1 TO 3: GOSUB 6500: NEXT J
6170  RETURN
6180 '
6190 FOR I=IP-1 TO IP+1
6200  FOR J=JP-1 TO JP+1
6210   GOSUB 6500
6220 NEXT J: NEXT I
6230 RETURN
6240 '
6500 '------------ SP INVERSION PION I,J ---------------------
6510 '
6520 IF I<1 OR I>3 OR J<1 OR J>3 THEN RETURN
6530 T(I,J)=-T(I,J)
6540 GOSUB 8500
6550 RETURN
6560 '
7000 '------------ SP LECTURE COUP PROPOSE -------------------
7010 '
7020 PEN 1: LOCATE 19,23
7030 PRINT "choisissez votre pion";
7040 I=1: J=1
7050 X=8*I+3: Y=6*J
7060 LOCATE X,Y: PRINT " ";
7070 IF T(I,J)=1 THEN PN=2 ELSE PN=3
7080 D$=INKEY$: IF D$="" THEN 7080
7090 IF D$=" " THEN 7170
7100 D=ASC(D$)
7110 I=I-(D=242)*(I>1)+(D=243)*(I<3)
7120 J=J-(D=240)*(J>1)+(D=241)*(J<3)
7130 PEN PN: LOCATE X,Y: PRINT P$;
7140 FOR K=1 TO 50: NEXT K
7150 GOTO 7050
7160 '
7170 SOUND 1,400,20
7180 PEN PN: LOCATE X,Y: PRINT P$;
7190 PEN 1: LOCATE 19,23: PRINT BL$;
7200 RETURN
7210 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 CLS
8030 PEN 1: LOCATE 11,2
8040 PRINT "F L I P - F L O P"
8050 NC=0
8060 NF=0
```

```
8070 FOR I=1 TO 3: FOR J=1 TO 3
8080  IF RND(1)>0.5 THEN T(I,J)=1 ELSE T(I,J)=-1
8090 NEXT J: NEXT I
8100 GOSUB 5000
8110 IF T(2,2)=1 AND NF=8 THEN 8060
8120 LOCATE 4,23
8130 PRINT "COUP  M-SCORE";
8140 LOCATE 12,24: PRINT MS;
8150 FOR I=1 TO 3: FOR J=1 TO 3
8160  GOSUB 8500
8170 NEXT J: NEXT I
8180 RETURN
8190 '
8500 '------------ SP DESSIN PION I,J -----------------------
8510 '
8520 XD=8*I+1: YD=6*J-2
8530 IF T(I,J)=1 THEN PEN 2 ELSE PEN 3
8540 FOR Y=YD TO YD+3
8550  LOCATE XD,Y: PRINT L$;
8560 NEXT Y
8570 RETURN
8580 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 MODE 1
9030 CE=0: CT=16: BR=8: CP=22: CF=20
9040 INK 0,CE: INK 1,CT: INK 2,CP: INK 3,CF
9050 PAPER 0: CLS
9060 '
9070 DIM T(3,3)
9080 MS=0
9090 P$=CHR$(143)
9100 L$="": FOR K=1 TO 4: L$=L$+P$: NEXT K
9110 BL$="": FOR K=1 TO 21: BL$=BL$+" ": NEXT K
9120 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur des pions**

- pour changer la couleur du côté pile (actuellement vert) : remplacez, en 9020, le nombre 22 par une valeur de votre choix (1 à 26),

- pour changer la couleur du côté face (actuellement bleu) : remplacez, en 9020, le nombre 20 par une valeur de votre choix (1 à 26).

▷ **Pour modifier la couleur du bord de l'écran (actuellement noir)**

Remplacez, en 9020, la valeur 0 (dans BR = 0) par un nombre de votre choix (1 à 26).

▷ **Pour modifier la couleur du fond (ici noir)**

Remplacez, en 9020, la valeur 0 (dans CE = 0) par un nombre de votre choix (1 à 26). Évitez cependant de choisir une couleur de fond identique à l'une des deux couleurs des pions (qui deviendraient alors invisibles). D'autre part, notez que la couleur du "curseur" est la même que la couleur de fond.

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard,

— Lecture du clavier,

— Examen du contenu de l'écran.

b) Le programme principal (100-260)

120 appelle le sous-programme d'initialisation du jeu.

130-250 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

140-170 correspondent au jeu d'un coup ; elles sont répétées jusqu'à ce que vous ayez gagné :

140-150 incrémentent le compteur de coups (NC) et affichent sa valeur.

160 appelle les sous-programmes : lecture du coup à jouer, jeu du coup (inversion des pions correspondants), analyse du coup.

170 examine si vous avez obtenu la situation gagnante.

180 met à jour le meilleur score.

190-210 vous affichent "*en musique*" que vous avez gagné.

220-240 vous demandent si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9110)

9030-9040 déterminent les différentes couleurs.

9050 efface l'écran et fixe la couleur de la bordure.

9070 réserve le tableau T(3,3) servant à représenter la situation (pile ou face) de chaque pion.

9100 prépare une chaîne de 4 pavés.

9110 prépare une chaîne de 21 caractères ''blanc''.

d) Le sous-programme d'initialisation d'une partie (8000-8180)

8020-8040 effacent l'écran et affichent ''FLIP FLOP''

8050 initialise le compteur de coups joués.

8060-8110 choisissent au hasard la position (pile ou face) de chaque pion en évitant la situation gagnante.

8120-8140 préparent l'affichage du score et affichent le meilleur score.

8150-8170 dessinent les pions à l'aide du sous-programme 8500.

e) Le sous-programme de dessin d'un pion (8500-8570)

Il détermine l'emplacement sur l'écran d'un pion donné (I et J contiennent sa position dans la grille 3 × 3). Il le dessine, en tenant compte de sa position (pile ou face) précisée dans le tableau T.

f) Le sous-programme de lecture du coup proposé (7000-7200)

7020-7030 vous affichent ''choisissez votre pion''.

7040-7150 vous permettent de sélectionner votre pion en employant les touches ↑ ↓ → ← ; elles sont répétées jusqu'à ce que vous frappiez la barre d'espace :

7040 initialise la position du curseur (pion en haut, à gauche).

7050-7060 affichent le curseur.

7070 ''mémorise'' dans PN la couleur du pion où se trouve le curseur.

7080-7090 attendent qu'une touche du clavier soit pressée et examinent s'il s'agit de la barre d'espace.

7100-7140 déterminent la nouvelle position du curseur et l'effacent de sa position actuelle (en le coloriant de la couleur du pion où il est situé, couleur préalablement mémorisée en PN).

7170-7190 émettent un son pour vous indiquer que votre coup a été pris en compte et effacent le curseur et le message ''choisissez...''.

g) Le sous-programme de jeu du coup proposé (6000-6230)

Lors de l'appel du sous-programme, I et J contiennent les coordonnées (dans la grille 3 × 3) du pion choisi. Dans tous les cas, le sous-programme 6500 est utilisé pour ''retourner'' les pions nécessaires.

6020-6090 cas du pion central.

6110-6130 cas d'un pion situé au centre d'une colonne (et non central).

6150-6170 cas d'un pion situé au centre d'une ligne (et non central).

6190-6230 cas d'un pion situé dans un coin.

h) Le sous-programme d'analyse d'un coup (5000-5060)

Il compte le nombre de pions placés côté ''face'' (bleue).

5. Liste des variables

CE	Couleur du fond.
CT	Couleur du ''titre'' (FLIP FLOP), des scores et des messages.
BR	Couleur du bord de l'écran.
CP	Couleur représentant le côté pile des pions.
CF	Couleur représentant le côté face des pions.
T(3,3)	Tableau représentant la situation de chaque pion : 1 : côté pile, — 1 : côté face.
MS	Meilleur score.
P$	Caractère représentant un pavé ''plein''
L$	Chaîne de 4 caractères utilisée pour représenter les pions.
BL$	Chaîne de 21 caractères ''blanc''.

NC	Numéro du coup.
NF	Nombre de pions côté face.
XD,YD	Coordonnées écran du coin haut gauche d'un pion.
I,J	Coordonnées d'un pion dans la grille (varient entre 1 et 3).
IP,JP	Coordonnées du pion choisi dans la grille (varient entre 1 et 3).
X,Y	Coordonnées du curseur.

Variables auxiliaires

R$ D$ D

5

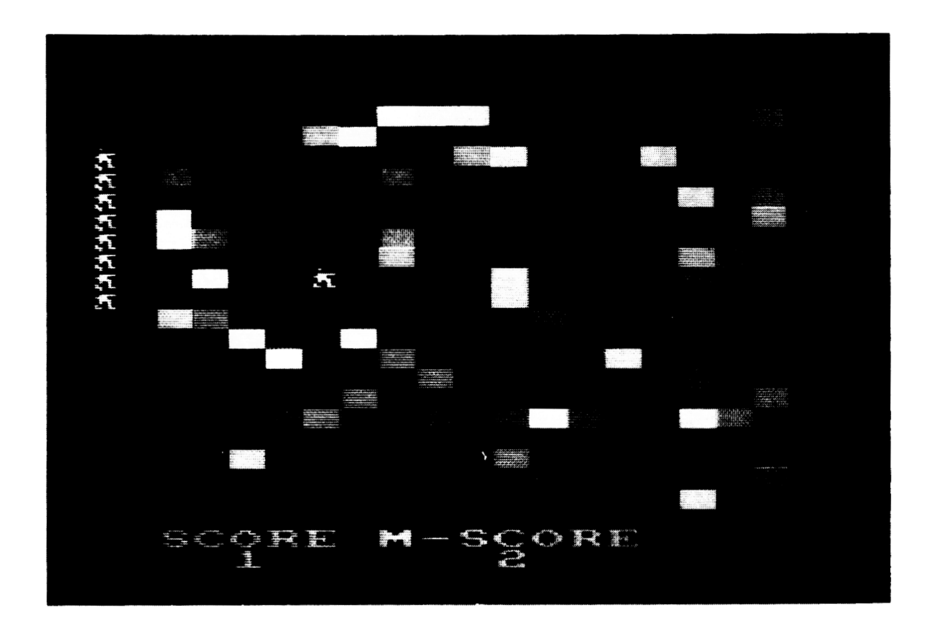
Drôles de mines

1. Le jeu

Ce sont effectivement de drôles de mines qu'il va vous falloir déjouer. En effet, contrairement aux mines habituelles, elles sont capables de ''sauter'' une multitude de fois.

Votre mission : faire parvenir un maximum d'espions en territoire ennemi. Pour cela, ils doivent obligatoirement traverser un domaine infesté de mines. Vous disposez de douze hommes, prêts à risquer leur vie, qui suivent à la lettre l'itinéraire que vous leur indiquez par radio.

Vos douze espions sont alignés sur la gauche de l'écran. Le premier attend que vous le dirigiez à l'aide des quatre touches fléchées (ou, le cas échéant, de la poignée de jeu). Vous devez tenter de le faire parvenir à droite de l'écran (à n'importe quelle hauteur). S'il saute sur une mine (avec sons et décors appropriés), vous verrez aussitôt l'espion suivant se présenter au départ.



Bien entendu, les mines sont invisibles. Par contre, de nombreux obstacles, infranchissables, vous serviront de repère.

Bonne chance ! et n'oubliez pas que ces mines sont drôles.

2. Le programme

```
100 '******************** DROLES DE MINES ********************
110 '
120 DEF FNSC(X,Y) = TEST(32*(X-1)+1,16*(25-Y)+1)
130 '
140 GOSUB 9000
150  GOSUB 8000
160  FOR IE=1 TO NE
170   X=XI: Y=YI: PEN 1
180   LOCATE X,Y: PRINT E$;
190   LOCATE X1-2,IE: PRINT " ";
200    D$=INKEY$: IF D$="" THEN 200
210    D=ASC(D$)
220    XN=X-(D=242)*(X>XI)+(D=243)*(X<XL)
230    YN=Y-(D=240)*(Y>Y1)+(D=241)*(Y<Y2)
```

```
240     P=FNSC(XN,YN)
250     IF P=4 THEN GOSUB 3000: GOTO 310
260     LOCATE X,Y: PRINT " ";
270     IF P=0 THEN X=XN: Y=YN
280     LOCATE X,Y: PRINT E$;
290     IF X=XL THEN GOSUB 2000: GOTO 310
300    GOTO 200
310   NEXT IE
320   GOSUB 1000
330 IF R$<>"N" THEN 150
340 END
350 '
1000 '------------ SP FIN DE PARTIE -------------------------
1010 '
1020 IF SC>MS THEN MS=SC
1030 PEN 3: LOCATE 18,22
1040 WHILE INKEY$<>"": WEND
1050 PRINT "rejouez vous (O/N)?";
1060 R$=INKEY$: IF R$="" THEN 1060
1070 RETURN
1080 '
2000 '------------ SP TRAVERSEE REUSSIE ---------------------
2010 '
2020 DATA 10
2030 DATA 3,426, 9,426, 12,426, 24,319, 24,319
2040 DATA 24,284, 24,284, 36,213, 12,253, 24,319
2050 '
2060 RESTORE 2020: READ N
2070 FOR I=1 TO N
2080  READ L,H: SOUND 1,H,3*L: SOUND 1,0,2
2090 NEXT I
2100 '
2110 SC=SC+1
2120 PEN 2: LOCATE 4,25: PRINT SC;
2130 LOCATE X,Y: PRINT " ";
2140 WHILE INKEY$<>"": WEND
2150 RETURN
2160 '
3000 '------------ SP EXPLOSION MINE ------------------------
3010 '
3020 PEN 1: LOCATE X,Y: PRINT " ";
3030 LOCATE XN,YN: PRINT E$;
3040 FOR K=1 TO 30: NEXT K
3050 LOCATE XN,YN: PRINT E1$;
```

```
3060 FOR I=1 TO 5
3070  FOR H=500 TO 50 STEP -20: SOUND 1,H,1: NEXT H
3080 NEXT I
3090 LOCATE XN,YN: PRINT E2$;
3100 FOR H=50 TO 2000 STEP 50: SOUND 1,H,1: NEXT H
3110 PEN 4: LOCATE XN,YN: PRINT M$;
3120 WHILE INKEY$<>"": WEND
3130 RETURN
3140 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 PAPER 0: CLS
8030 SC=0
8040 PEN 2
8050 LOCATE 3,24: PRINT "SCORE M-SCORE";
8060 LOCATE 11,25: PRINT MS;
8070 '
8080 FOR I=1 TO NO
8090  X=X1+INT(RND(1)*(X2-X1+1))
8100  Y=Y1+INT(RND(1)*(Y2-Y1+1))
8110  IF FNSC(X,Y)<>0 THEN 8090
8120  C=INT(5+11*RND(1)): PEN C
8130  LOCATE X,Y: PRINT O$;
8140 NEXT I
8150 '
8160 FOR I=1 TO NM
8170  X=X1+INT(RND(1)*(X2-X1+1))
8180  Y=Y1+INT(RND(1)*(Y2-Y1+1))
8190  IF FNSC(X,Y)<>0 THEN 8170
8200  PEN 4: LOCATE X,Y: PRINT M$;
8210 NEXT I
8220 '
8230 PEN 1
8240 FOR I=1 TO NE
8250  LOCATE X1-2,I: PRINT E$;
8260 NEXT I
8270 RETURN
8280 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 MODE 0
9030 CF=0: CE=24: BR=0: CS=16: CD=14
9040 INK 0,CF: INK 1,CE: INK 2,CS: INK 3,CD: INK 4,CF
9050 CLS: BORDER BR
9060 '
```

```
9070 DATA 2,4,5,6,7,8,10,12,15,15,18
9080 DIM C(16)
9090 RESTORE 9070
9100 FOR I=5 TO 15: READ C(I): INK I,C(I): NEXT I
9110 '
9120 X1=3: X2=19: XL=20
9130 Y1=3: Y2=22: XI=2: YI=15
9140 NE=12: NO=65: NM=20
9150 '
9160 SYMBOL AFTER 160
9170 SYMBOL 160,16,0,30,56,28,20,36,54
9180 SYMBOL 161, 40,46,0,24,16,4,68,118
9190 SYMBOL 162,4,32,2,8,64,2,40,2
9200 E$=CHR$(160): E1$=CHR$(161): E2$=CHR$(162)
9210 M$=CHR$(143): O$=CHR$(143)
9220 '
9230 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier le nombre d'espions**

Remplacez le nombre 12 de l'instruction 9130 par une valeur de votre choix ne dépassant pas 22.

▷ **Pour modifier le nombre d'obstacles**

Remplacez le nombre 65 de l'instruction 9130 par la valeur de votre choix.

▷ **Pour modifier le nombre de mines**

Remplacez le nombre 20 de l'instruction 9130 par la valeur de votre choix.

Notez bien qu'un trop grand nombre de mines risque de rendre le jeu difficile, voir impossible. Un trop petit nombre risque, par contre, de le rendre trop facile.

▷ **Pour que les mines n'explosent qu'une seule fois**

Les mines ne seront plus ''drôles'' si vous remplacez la ligne 3110 par :

```
3110 PEN 4 : LOCATE XN,YN : PRINT "  " ;
```

▷ **Pour utiliser une poignée de jeu**

Remplacez les lignes 200 à 230 par :

```
200 D=JOY (0) : IF D=0 THEN 200
210 FOR K=1 TO 50 : NEXT K
220 XN=X-(D=4) * (X>XI) + (D=8) * (X<XL)
230 YN=Y-(D=1) * (Y>Y1) + (D=2) * (Y<Y2)
```

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard,

— Animation,

— Lecteur du clavier,

— Examen du contenu de l'écran.

b) Le programme principal (100-340)

120 définit une fonction permettant de connaître la couleur (numéro d'encrier) d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

140 appelle le sous-programme d'initialisation du jeu.

150-330 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne voulez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

150-310 répètent douze fois la tentative de traversée d'un espion. Pour chaque tentative :

170-190 initialisent les coordonnées (X,Y) de l'espion à la position de départ, y dessinent un espion et en effacent un de la file d'attente.

190-300 sont répétées jusqu'à ce que l'espion ait sauté sur une mine ou qu'il ait réussi sa traversée :

200-230 attendent qu'une touche soit enfoncée et déterminent la nouvelle position correspondante (XN,YN).

240 détermine le contenu de cette nouvelle position (voir annexe : examen du contenu de l'écran).

250-280 appellent le sous-programme ''explosion mine'' lorsqu'une mine se trouve à cette nouvelle position. Dans le cas contraire, et il ne s'agit pas d'un obstacle, l'espion est dessiné dans sa nouvelle position après avoir été effacé de l'ancienne.

290 appelle le sous-programme ''traversée réussie'' si l'espion a atteint le bord droit.

320 appelle le sous-programme de fin de partie (qui actualise le meilleur score et vous demande si vous souhaitez rejouer).

330 examine votre réponse.

c) Le sous-programme d'initialisation du jeu (9000-9230)

9020-9050 définissent les différentes couleurs et effacent l'écran.

9070-9100 placent dans les ''encriers'' 5 à 15 les couleurs qui seront utilisées pour représenter les obstacles.

9120-9130 fixent les limites de jeu et la position initiale d'un espion.

9140 fixe le nombre d'espions, le nombre d'obstacles et le nombre de mines.

9160-9210 fabriquent les différents caractères graphiques.

d) Le sous-programme d'initialisation d'une partie (8000-8270)

8020-8060 effacent l'écran, initialisent le score et affichent le meilleur score.

8080-8140 dessinent les différents obstacles en choisissant au hasard leur couleur et leur position (8110 vérifie que l'emplacement choisi est libre).

8160-8210 placent les mines au hasard (8190 vérifie que l'emplacement choisi est disponible).

REMARQUE : les mines sont en fait dessinées dans la même couleur que le fond et ceci afin d'être invisibles. Elles restent néanmoins détectables par la fonction TEST. En effet, elles sont réalisées avec un numéro d'encre (4) différent de celui du fond (O), bien que correspondant à la même couleur (ici O).

8230-8270 dessinent la file d'attente des espions.

e) Le sous-programme d'explosion d'une mine (3000-3130)

3020-3080 dessinent d'abord l'espion à l'emplacement voulu puis, après un court instant, en font apparaître ses ''morceaux'' et émettent une succession de sons appropriés.

3090-3100 simulent la destruction complète de l'espion en faisant apparaître ses "miettes" et un son suggestif.

3110 replace la mine à l'emplacement du sinistre.

f) Le sous-programme "traversée réussie" (2000-2160)

2060-2090 font retentir un air de victoire (ressemblant à "la marseillaise").

2110-2120 incrémentent le score et l'affichent.

2130 efface l'espion qui vient de réussir sa traversée.

g) Le sous-programme de fin de partie (1000-1070)

1020 actualise le meilleur score.

1030-1060 vous demandent si vous souhaitez rejouer.

5. Liste des variables

CF	Couleur du fond.
CE	Couleur de l'espion.
BR	Couleur du tour.
CS	Couleur d'affichage des scores.
CD	Couleur d'affichage des messages.
X1,X2,Y1,Y2	Coordonnées du domaine dans lequel peuvent apparaître les mines et les obstacles.
XL	Abscisse à atteindre par l'espion pour que sa traversée soit considérée comme réussie.
XI,YI	Coordonnées de la position de départ d'un espion.
NE	Nombre d'espions.
NO	Nombre d'obstacles
NM	Nombre de mines.
E$	Caractère graphique représentant l'espion (en vie).
E1$,E2$	Caractères graphiques représentant l'espion (ou plutôt ses restes) lors d'une explosion.

M$	Caractère représentant une mine. Il s'agit d'un carré plein.
SC	Score (nombre d'espions ayant réussi leur traversée).
IE	Numéro de l'espion en cours.
SC	Coordonnées de l'espion en cours de traversée.
XN,YN	Coordonnées du prochain emplacement de l'espion.
MS	Meilleur score.

Variables auxiliaires

R$ A C D$ D P I

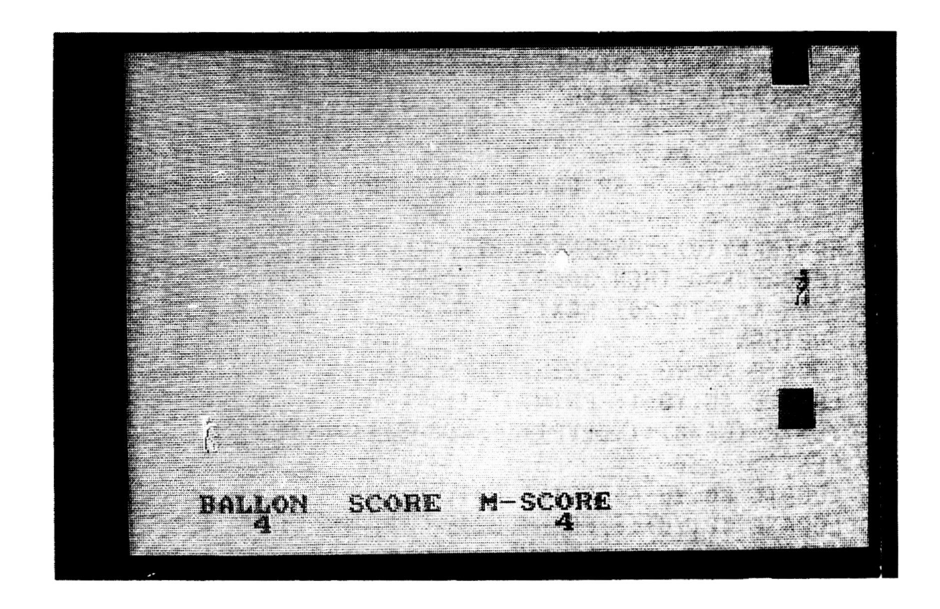
X,Y (dans le sous-programme 8000 uniquement).

6
GOAL

1. Le jeu

Vous venez d'être sélectionné comme goal dans une grande équipe de football. Saurez-vous être à la hauteur de la situation en interceptant tous les tirs de vos adversaires qui se montrent particulièrement adroits.

Les buts sont matérialisés par deux poteaux sur la droite de l'écran. Vous déplacez le goal, entre les deux poteaux à l'aide des deux touches " ↑ " et " ↓ ". Douze tirs au but vont avoir lieu. Pour chaque tir, vous voyez apparaître en un endroit quelconque de l'écran, un joueur accompagné d'un ballon. Au bout d'un court instant (variable d'une fois à l'autre), le joueur "shoote" dans le ballon et il vous faut l'intercepter. Vous pouvez déplacer le goal dès que le joueur apparaît. Comme vous le constaterez, vos adversaires sont très habiles puisqu'ils tirent toujours dans les buts.



Le programme affiche le nombre d'essais, le nombre de ballons interceptés et le meilleur score réalisé depuis le début du jeu.

2. Le programme

```
100 '********************** GOAL ******************************
110 '
120 DEF FNSC(X,Y)=TEST(16*(X-1)+7,16*(25-Y)+13)
130 GOSUB 9000
140  GOSUB 8000
150  FOR IC=1 TO NC
160   GOSUB 5000
170    GOSUB 3000
180    LOCATE X,Y: PRINT " ";
190    X=X+2: Y=Y+DY
200    IF X>=XG THEN GOSUB 4000: GOTO 230
210    PEN 3: LOCATE X,Y: PRINT B$;
220   GOTO 170
230  NEXT IC
240  IF SC>MS THEN MS=SC
250  PEN 3: LOCATE 6,13
260  WHILE INKEY$<>"": WEND
```

```
270  PRINT "voulez vous rejouer (O/N)?";
280  R$=INKEY$: IF R$="" THEN 280
290 IF R$<>"N" THEN 140
300 END
310 '
3000 '------------ SP DEPLACEMENT GOAL ----------------------
3010 '
3020 GA=INKEY(0): DR=INKEY(2)
3030 IF GA*DR=0 THEN 3060
3040  FOR K=1 TO 20: NEXT K
3050  RETURN
3060 LOCATE XG,YG  : PRINT " ";
3070 LOCATE XG,YG+1: PRINT " ";
3080 YG=YG-(GA=0)*(YG>YP1+1)+(DR=0)*(YG<YP2-2)
3090 PEN 1
3100 LOCATE XG,YG  : PRINT GH$;
3110 LOCATE XG,YG+1: PRINT GB$;
3120 RETURN
3130 '
4000 '---------- SP FIN DE COUP -----------------------------
4010 '
4020 IF FNSC(X,Y)=0 AND FNSC(X-1,Y)=0 THEN 4070
4030  SC=SC+1
4040  SOUND 1,379,6
4050  PEN 2: LOCATE 15,24: PRINT SC;
4060 GOTO 4090
4070  FOR H=500 TO 200 STEP -20: SOUND 1,H,1: NEXT H
4080 '
4090 LOCATE XJ,YJ:   PRINT " ";
4100 LOCATE XJ,YJ+1: PRINT " ";
4110 FOR K=1 TO 100: NEXT K
4120 RETURN
4130 '
5000 '------------ SP INITIALISATION D'UN COUP --------------
5010 '
5020 PEN 2: LOCATE 7,24: PRINT IC;
5030 '
5040  X=X1+2*INT(RND(1)*(X2-X1+1)/2)
5050  Y=Y1+2*INT(RND(1)*(Y2-Y1+1)/2)
5060  DY=INT(RND(1)*5)-2
5070  YI=Y+(XP-X+1)*DY/2
5080 IF YI<=YP1+1 OR YI>YP2-1 THEN 5040
5090 '
5100 XJ=X-1: YJ=Y-1
5110 PEN 3: LOCATE X,Y: PRINT B$;
```

```
5120 LOCATE XJ,YJ  : PRINT JH$;
5130 LOCATE XJ,YJ+1: PRINT JB1$;
5140 '
5150 FOR IA=1 TO TM*RND(1)
5160  GOSUB 3000
5170 NEXT IA
5180 '
5190 PEN 3: LOCATE XJ,YJ+1: PRINT JB2$;
5200 SOUND 1,478,6
5210 RETURN
5220 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 SC=0: XG=XP: YG=(YP1+YP2)/2: CLS
8030 '
8040 PEN 2
8050 LOCATE XP-1,YP1-1: PRINT P$;
8060 LOCATE XP-1,YP1  : PRINT P$;
8070 LOCATE XP-1,YP2  : PRINT P$;
8080 LOCATE XP-1,YP2+1: PRINT P$;
8090 '
8100 PEN 1
8110 LOCATE XG,YG  : PRINT GH$;
8120 LOCATE XG,YG+1: PRINT GB$;
8130 '
8140 PEN 2: LOCATE 5,23
8150 PRINT "BALLON  SCORE  M-SCORE";
8160 LOCATE 23,24: PRINT MS;
8170 RETURN
8180 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 MODE 1
9030 CE=21: BR=0: CG=6 : CP=7: CB=2
9040 INK 0,CE: INK 1,CG: INK 2,CP: INK 3,CB
9050 BORDER BR: PAPER 0: CLS
9060 '
9070 X1=4: X2=26: Y1=2: Y2=21
9080 YP1=2: YP2=18: XP=37
9090 NC=12: TM=100: MS=0
9100 '
9110 SYMBOL AFTER 160
9120 SYMBOL 160, 0,56,56,48,63,56,52,50
9130 SYMBOL 161, 49,40,40,40,40,40,60,0
```

```
9140 SYMBOL 162, 49,40,36,38,38,38,51,0
9150 SYMBOL 163, 0,28,28,12,252,28,44,76
9160 SYMBOL 164, 120,20,36,68,68,68,204,0
9170 SYMBOL 165, 24,60,126,126,126,126,60,24
9180 JH$=CHR$(160): JB1$=CHR$(161): JB2$=CHR$(162)
9190 GH$=CHR$(163): GB$=CHR$(164): B$=CHR$(165)
9200 P$=CHR$(143)+CHR$(143)
9210 '
9220 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

▷ **Pour changer les couleurs**

- *du terrain*

Remplacez, en 9030, la valeur 21 (dans CE = 21) par un nombre de votre choix (entre 0 et 26).

- *du tour de l'écran*

Remplacez, en 9030, la valeur 0 (dans BR = 0) par un nombre de votre choix (entre 0 et 26).

- *du joueur*

Remplacez, en 9030, la valeur 2 (dans CB = 2) par un nombre de votre choix (entre 0 et 26).

- *du goal*

Remplacez, en 9030, la valeur 6 (dans CG = 6) par un nombre de votre choix (entre 0 et 26).

▷ **Pour modifier le nombre d'essais d'une partie**

Remplacez, en 9090, le nombre 12 (dans NC = 12) par une valeur de votre choix.

▷ **Pour modifier la taille des buts**

Modifiez, en 9080, les valeurs de YP1 (ici 2) et de YP2 (ici 18) qui fixent la position des poteaux.

Ainsi par exemple :

```
9080 YP1=8: YP2=18 : XP=37
```

vous fourniront des petits buts, tandis qu'avec :

```
9080 YP1=2 : YP2=22 : XP=37
```

ils occuperont presque toute la hauteur de l'écran.

▷ **Pour utiliser une poignée de jeu**

Remplacez les lignes 3020 et 3030 par :

```
3020 D=JOY(0)
3030 IF D <>0 THEN 3060
```

et remplacez 3080 par :

```
3080 YG=YG-(D=1) * (YG >YP1+1) + (D=2) * (YG<YP2-2)
```

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard,

— Animation,

— Lecture rapide du clavier,

— Examen du contenu de l'écran.

b) Le programme principal (100-300)

120 définit une fonction permettant de connaître le numéro d'encre d'*un* point graphique de l'emplacement texte de coordonnées X,Y (ce point est choisi de telle manière qu'il figure à la fois dans la partie haute et la partie basse du goal).

130 appelle le sous-progamme d'initialisation du jeu.

140-290 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150-230 proposent les douze essais. Pour chaque coup :

160 appelle le sous-programme de début d'un coup.

170-220 sont répétées jusqu'à ce que le ballon atteigne la ligne des buts :
170 appelle le sous-programme de déplacement du goal.
180-210 déplacent le ballon et vérifient s'il atteint la ligne des buts.

240 actualise le meilleur score.

250-280 vous demandent si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9220)

9020-9050 déterminent les différentes couleurs.

9070-9080 fixent les limites de la zone à l'intérieur de laquelle le joueur peut apparaître ainsi que la position des poteaux.

9090 fixe le nombre d'essais, le temps d'attente maximum et initialise le meilleur score.

9110-9200 fabriquent les caractères graphiques utilisés pour représenter le joueur, le goal et le ballon.

d) Le sous-programme d'initialisation d'une partie (8000-8170)

8030 initialise le score, détermine la position initiale du goal et efface l'écran.

8040-8080 dessinent les poteaux des buts.

8100-8120 dessinent le goal dans sa position initiale.

8140-8170 affichent le meilleur score.

e) Le sous-programme d'initialisation d'un coup (5000-5210)

5020 affiche le numéro du coup qui va être joué.

5040-5050 choisissent au hasard un emplacement pour le joueur.

5060-5080 choisissent au hasard une direction de tir telle que la trajectoire du ballon passe entre les deux poteaux.

5100-5130 dessinent le ballon et le joueur dans la position "prêt à shooter".

5150-5170 permettent d'attendre pendant un temps tiré au hasard. La répétition de l'appel du sous-programme 3000 permet de déplacer le goal pendant ce temps.

5190-5210 dessinent le joueur en train de shooter dans le ballon et émettent un son approprié.

f) Le sous-programme de déplacement du goal (3000-3120)

3020-3050 examinent le clavier. Si aucune des 2 touches ← et → n'est pressée, on quitte le sous-programme après un léger temps d'attente (qui permet ainsi au ballon de ne pas se déplacer plus vite que lorsque vous déplacez le goal).

3060-3070 effacent le goal de sa position actuelle.

3080-3110 déterminent la nouvelle position du goal et l'y redessinent.

g) Le sous-programme de fin d'un coup (4000-4130)

4020 examine si le ballon est intercepté par le goal.

4030-4060 exécutées en cas de succès, ces instructions actualisent le score, l'affichent et émettent un son approprié.

4070 émet un son correspondant à l'échec.

4090-4110 effacent le joueur et attendent un instant.

5. Liste des variables

CE	Couleur du terrain.
BR	Couleur du pourtour de l'écran.
CG	Couleur du goal.
CP	Couleur des poteaux des buts et des scores.
CB	Couleur du joueur et du ballon.
X1,X2,Y1,Y2	Coordonnées du domaine dans lequel le joueur peut apparaître.
XP	Abscisse de l'extrémité droite des poteaux.
YP1	Ordonnée de l'extrémité basse du poteau supérieur.
YP2	Ordonnée de l'extrémité haute du poteau inférieur.
NC	Nombre de coups d'une partie.
TM	Temps d'attente maximum s'écoulant entre l'apparition du joueur et le moment où il shoote dans le ballon.
MS	Meilleur score.
JH$	Caractère graphique représentant la partie supérieure du joueur.
JB1$	Caractère graphique représentant la partie inférieure du joueur en position de repos.
JB2$	Caractère graphique représentant la partie inférieure du joueur lorsqu'il shoote dans le ballon.
GH$	Caractère graphique représentant la partie supérieure du goal.
GB$	Caractère graphique représentant la partie inférieure du goal.
SC	Score (nombre de ballons interceptés).

XG,YG	Coordonnées du goal.
IC	Numéro du coup qui est en train de se jouer.
X,Y	Coordonnées du ballon.
DY	Indique la direction de déplacement du ballon. Sa valeur (– 2, – 1, 0, 1 ou 2) représente le déplacement vertical correspondant à un déplacement horizontal de deux unités.
YI	Ordonnée de l'intersection de la trajectoire du ballon avec la ligne des buts.
XJ,YJ	Coordonnées du joueur.

Variables auxiliaires

R$ A R IA

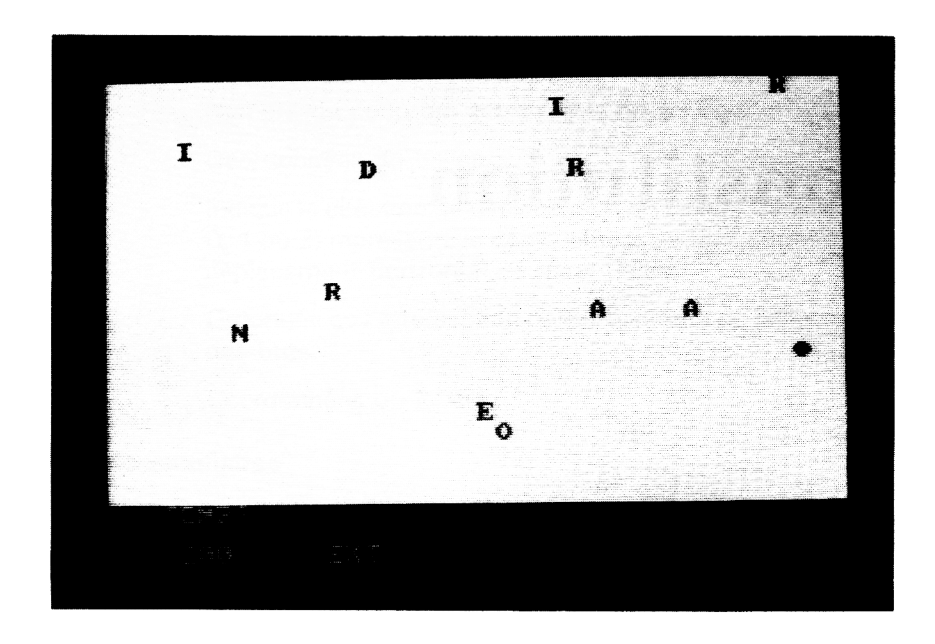
7
Reconstitution

1. Le jeu

Un crime vient d'être commis. La victime a été coupée en morceaux ! Et le pire, c'est que vous êtes chargé de la reconstitution. De la reconstitution des circonstances du crime, pensez-vous ! Vous n'y êtes pas... il s'agit de la reconstitution de la victime...

Dure affaire, pensez-vous. En fait, nous jouons sur les mots puisque la victime n'est autre qu'un mot qui va se présenter à vous avant qu'une explosion n'en disperse les lettres sur l'écran. Vote mission consiste à les ramasser le plus vite possible et dans le bon ordre.

Vous ramassez les lettres grâce à une "boule" que vous promenez sur l'écran à l'aide des quatre touches fléchées. Au début de la partie, le programme vous laisse examiner à loisir l'état de la situation : position des lettres, de la boule, meilleur chemin à suivre... Quand vous pensez être prêt, vous appuyez sur une tou-



che du clavier... A partir de là, dès que vous aurez mis votre boule en mouvement en la dirigeant à l'aide de l'une des quatre flèches, plus rien ne pourra l'arrêter.

Notez bien que pour votre boule, l'écran est ''sphérique''. Autrement dit, si la boule disparaît à gauche, elle réapparaît aussitôt à droite au même niveau. De même, si elle disparaît en haut, elle réapparaît en bas de l'écran, au-dessus des trois lignes servant à l'affichage d'informations.

Au fur et à mesure que vous ramassez les lettres, vous voyez s'inscrire, en bas de l'écran le mot partiellement reconstitué. Le programme affiche en permanence le temps écoulé.

Si vous gagnez, vous voyez apparaître le mot ''GAGNÉ'' et votre temps moyen (par lettre) ainsi que le meilleur score. Si vous perdez, vous ne verrez que le mot ''PERDU''.

2. Le programme

```
100 '****************** RECONSTITUTION ***********************
110 '
120 GOSUB 9000
130  GOSUB 8000: OK=0: RESTORE 8380
140  FOR NL=1 TO LEN(MO$)
150    R$=INKEY$: IF R$="" THEN DN=0 ELSE DN=ASC(R$)
160    IF DN>=240 AND DN<=243 THEN DL=DN
170    XN=X+(DL=242)-(DL=243)
180    YN=Y+(DL=240)-(DL=241)
190    IF XN<X1 THEN XN=X2
200    IF XN>X2 THEN XN=X1
210    IF YN<Y1 THEN YN=Y2
220    IF YN>Y2 THEN YN=Y1
230    T=T+1: LOCATE #1,4,3: PRINT #1, T;
240    IF DL=0 THEN 150
250    C$=EC$(XN,YN)
260    LOCATE X,Y: PRINT " ";
270    X=XN: Y=YN: LOCATE X,Y: PRINT B$;
280   IF C$="" THEN 150
290   IF C$<>MID$(MO$,NL,1) THEN 350
300   LOCATE #1,13,3: PRINT #1,LEFT$(MO$,NL)
310   EC$(X,Y)=""
320   READ H: SOUND 1,H,10
330  NEXT NL
340  OK=1
350  GOSUB 7000
360 IF R$<>"N" THEN 130
370 END
380 '
7000 '------------ SP FIN DE PARTIE -------------------------
7010 '
7020 FOR K=1 TO 200: NEXT K
7030 CLS #1
7040 IF OK=1 THEN 7100
7050 '
7060  LOCATE #1,2,2: PRINT #1,"PERDU";
7070  FOR H=100 TO 700 STEP 10: SOUND 1,H,1: NEXT H
7080 GOTO 7160
7090 '
7100  LOCATE #1,2,2: PRINT #1,"GAGNE";
7110  FOR H=250 TO 190 STEP -20: SOUND 1,H,20: NEXT H
7120  TM=INT(T/NL)
```

```
7130  LOCATE #1,13,1: PRINT #1,"TEMPS MOYEN";TM;
7140  LOCATE #1,13,2: PRINT #1,"MEILLEUR TEMPS"; MT;
7150 '
7160 WHILE INKEY$<>"": WEND
7170 LOCATE #1,13,4: PRINT #1,"rejouez vous (O/N)?";
7180 R$=INKEY$: IF R$="" THEN 7180
7190 RETURN
7200 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 CLS: CLS #1
8030 '
8040 T=0
8050 K=1+INT(RND(1)*NM): MO$=MO$(K)
8060 LOCATE 10,10: PRINT MO$;
8070 FOR X=1 TO X2: FOR Y=1 TO Y2
8080  EC$(X,Y)=""
8090 NEXT Y: NEXT X
8100 '
8110 RESTORE 8380
8120 FOR I=1 TO LEN(MO$)
8130  X=X1+INT(RND(1)*(X2-X1+1))
8140  Y=Y1+INT(RND(1)*(Y2-Y1+1))
8150  IF EC$(X,Y)<>"" OR Y=10 THEN 8130
8160  LOCATE 10+I-1,10: PRINT " ";
8170  READ H: SOUND 1,H,10
8180  LOCATE X,Y: PRINT MID$(MO$,I,1);
8190  EC$(X,Y)=MID$(MO$,I,1)
8200 FOR K=1 TO 100: NEXT K
8210 NEXT I
8220 '
8230 X=X1+INT(RND(1)*(X2-X1+1))
8240 Y=Y1+INT(RND(1)*(Y2-Y1+1))
8250 IF EC$(X,Y)<>"" THEN 8230
8260 LOCATE X,Y: PRINT B$;
8270 DL=0
8280 '
8290 WHILE INKEY$<>"": WEND
8300 LOCATE #1, 2,2
8310 PRINT #1, "pour commencer, tapez sur une touche";
8320 R$=INKEY$: IF R$="" THEN 8320
8330 '
8340 CLS #1
8350 LOCATE #1, 4,1: PRINT #1, "TEMPS";
8360 RETURN
```

```
8370 '
8380 DATA 478,426,379,358,319,284,253
8390 DATA 239,213,190,179,159,142,127
8400 DATA 119,106,95,89,80,71,63
8410 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 CE1=23: CE2=1: IN=6: CM=24: BR=0
9030 INK 0,CE1: INK 1,IN: INK 2,CE2: INK 3,CM
9040 PAPER 0:BORDER BR:CLS
9050 '
9060 MT=0
9070 X1=1: X2=40: Y1=1: Y2=21
9080 DIM EC$(X2,Y2)
9090 B$=CHR$(231)
9100 '
9110 DATA 9
9120 DATA EXTRAORDINAIRE, MAGNIFIQUEMENT, OSTENSIBLEMENT
9130 DATA PARCIMONIEUSEMENT,  DECALCOMANIE, CARACTERISATION
9140 DATA BUDGETISATION, ENREGISTREMENT, PROGRAMMATION
9150 '
9160 RESTORE 9110
9170 READ NM: DIM MO$(NM)
9180 FOR I=1 TO NM
9190  READ MO$(I)
9200 NEXT I
9210 '
9220 WINDOW #1, 1,40,Y2+1,25
9230 PAPER #1,2
9240 PEN #1,3
9250 CLS #1
9260 PAPER 0: PEN 1
9270 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur du fond**

Remplacez, en 9020, la valeur 23 (dans CE1 = 23) par un nombre de votre choix (entre 0 et 26).

▷ **Pour modifier la couleur de la boule et des lettres**

Remplacez, en 9020, la valeur 6 (dans IN = 6) par un nombre de votre choix (entre 0 et 26).

▷ **Pour jouer sur un domaine plus petit**

Agissez sur la ligne 9070 en augmentant les valeurs de X1 et Y1 et en diminuant celles de X2 et Y2. Vous pouvez essayer, par exemple :

```
9070 X1=10 : X2=30 : Y1=5 : Y2=20
```

▷ **Pour ajouter de nouveaux mots ou pour modifier les mots proposés**

Il vous faut savoir que les instructions "DATA" des lignes 9110 à 9140 fournissent :

- le nombre total de mots (ici 9),
- la liste de ces mots. Chaque instruction peut contenir un nombre quelconque de mots ; ceux-ci doivent simplement être séparés par des virgules.

Notez bien que vous pouvez placer, non seulement des mots ayant un sens, mais éventuellement n'importe quelles suites de caractères (lettres, chiffres...). Cela peut rendre le jeu quelque peu plus compliqué.

▷ **Pour utiliser une poignée de jeu**

Remplacez les lignes 150 et 180 par :

```
150 DN=JOY(0)
160 IF DN=1 OR DN=2 OR DN=4 OR DN=8 THEN DL=DN
170 XN=X+(DL=4) - (DL=8)
180 YN=Y + (DL=1) - (DL=2)
```

4. Description du programme

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Lecture du clavier,
- Examen du contenu de l'écran

b) Le programme principal (100-370)

120 appelle le sous-programme d'initialisation du jeu.

130-360 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne voulez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie et place le pointeur de "data" sur le début des instructions "DATA" contenant les hauteurs des notes qui seront jouées au fur et à mesure que vous reconstituerez le mot voulu. (Notez que l'instruction RESTORE est nécessaire ici, car d'autres instructions DATA sont lues par ailleurs).

140-330 répètent les instructions 150-320 (correspondant au ramassage d'une lettre) jusqu'à ce que toutes les lettres aient été ramassées ou que vous en ratiez une. Pour chaque lettre :

170-290 sont répétées jusqu'à ce que la boule rencontre une lettre (bonne ou mauvaise) :

150-160 déterminent, dans DL, le code du nouveau déplacement de la boule, en fonction de l'ancien (DN) ou de la touche enfoncée.

170-220 déterminent les nouvelles coordonnées de la boule, en tenant compte de ce que notre domaine est sphérique.

230 actualise le temps et l'affiche.

240 s'assure que la boule n'est pas immobile (cas du début d'une partie).

250-270 déplacent la boule après avoir mémorisé dans C$, le caractère se trouvant au nouvel emplacement.

280 examine si le nouvel emplacement était disponible.

290 examine si la lettre rencontrée est la bonne.

300-320 affichent la lettre rencontrée, lorsqu'elle est correcte et émettent la note suivante (de la liste fournie en DATA).

340 exécutée uniquement lorsque toutes les lettres ont été correctement ramassées, cette instruction positionne l'indicateur OK.

350 appelle (dans tous les cas) le sous-programme de fin de partie.

360 examine si vous voulez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9270)

9020-9040 déterminent les différentes couleurs.

9060 initialise le meilleur score.

9070 fixe les limites du domaine où évolue la boule.

9080 réserve un tableau où seront conservés les caractères affichés en chaque emplacement de l'écran.

9090 fabrique le caractère ''boule''.

9110-9200 remplissent, grâce aux ''DATA'' des lignes 9110 à 9140, le tableau MO$ dans lequel un mot sera tiré au hasard à chaque partie.

9220-9250 définissent l'emplacement et les couleurs de la fenêtre numéro 1 utilisée pour les différents affichages en bas de l'écran.

d) Le sous-programme d'initialisation d'une partie (8000-8410)

8020 efface l'écran.

8040 initialise le compteur de temps.

8060-8070 choisissent un mot au hasard (dans MO$) et l'affichent.

8070-8090 initialisent le tableau EC$.

8110-8210 affichent en musique chaque lettre du mot en un emplacement quelconque de l'écran. Pour chaque lettre, on lit la ''note'' à jouer dans les ''DATA'' des lignes 8380 à 8400.

8230-8260 déterminent au hasard un emplacement disponible pour la boule et la dessinent.

8270 initialise le déplacement de la boule.

8290-8320 attendent que vous frappiez sur une touche.

8290-8250 effacent la partie inférieure de l'écran et affichent le titre ''TEMPS''.

e) Le sous-programme de fin de partie (7000-7190)

7020-7030 attendent un instant avant d'effacer la partie inférieure de l'écran.

7040 examine si vous avez gagné (OK = 1).

7060-7080 affichent le mot ''PERDU'' et émettent un son approprié, en cas d'échec.

7100-7140 affichent, en cas de réussite, le mot ''GAGNÉ'' en émettant un son approprié, affichent le temps moyen et le meilleur score et actualisent le meilleur score.

7160-7180 vous demandent si vous voulez rejouer.

5. Liste des variables

CE1	Couleur du domaine où évolue la boule.
CE2	Couleur de la zone inférieure de l'écran.
IN	Couleur de la boule et des lettres.
BR	Couleur du tour de l'écran.
CM	Couleur d'affichage du mot partiellement reconstitué, du temps et des scores.
MT	Meilleur temps moyen réalisé au cours des parties précédentes.
X1,X2,Y1,Y2	Coordonnées du domaine de jeu.
EC$(X2,Y2)	Tableau utilisé pour garder une trace des caractères affichés en chacun des emplacements de l'écran.
B$	Caractère graphique représentant la boule.
NM	Nombre total de mots parmi lesquels on en choisira un au hasard
MO$(NM)	Tableau contenant les différents mots parmi lesquels on en tirera un au hasard.
T	Temps écoulé depuis le début de la partie.
MO$	Mot tiré au hasard pour la partie en cours.
DL	Sens de déplacement de la boule. O : boule immobile (en début de partie seulement) 240 : haut 242 : gauche 241 : bas 243 : droite
OK	Indicateur ''gagné/perdu''. Initialisé en 0 en début de partie, cet indicateur est placé à 1 lorsque le joueur a reconstitué le mot.
NL	Numéro de la prochaine lettre à ''ramasser''.
X,Y	Coordonnées courantes de la boule.
XN,YN	Coordonnées du prochain emplacement de la boule.
SX	Contenu du prochain emplacement de la boule.
TM	Temps moyen réalisé au cours de la partie.

Variables auxiliaires

I   R$ A   K

X et Y (dans le sous-programme 8000)

H   DN

8
Phosphore

1. Le jeu

Le but de ce jeu de mémoire consiste à retenir des suites de lettres les plus longues possibles. Dix essais vous sont proposés. Pour chacun d'entre eux, le programme commence par vous montrer pendant un court instant une première lettre choisie au hasard. Il vous demande ensuite quelle était la lettre affichée. Si votre réponse est correcte, le programme vous gratifie d'un petit compliment et vous montre à nouveau cette lettre suivie d'une deuxième. Cela se poursuit ainsi avec trois lettres, puis quatre... jusqu'à ce que vous vous trompiez. Vous marquez alors un nombre de points égal au nombre de lettres que vous avez réussi à retenir.

Le programme affiche en permanence le numéro de l'essai que vous êtes en train de jouer, le nombre de lettres que vous avez déjà mémorisées pour l'essai en cours, votre score et le meilleur score.

Chaque lettre proposée est accompagnée d'une note de musique (qui reste la même pour une lettre donnée) ; cela peut quelque peu faciliter votre mémorisation.

2. Le programme

```
100 '****************** PHOSPHORE ****************************
110 '
120 GOSUB 9000
130  GOSUB 8000
140  FOR CC=1 TO NC
150   LE$=""
160   LOCATE 12,2: PRINT CC;
170   GOSUB 6000: GOSUB 6500: GOSUB 5000
180   IF OK=1 THEN 170
190   GOSUB 7500
200  NEXT CC
210  IF SC>MS THEN MS=SC
220  CLS#1: INK 2,CD: INK 3,CED
230  LOCATE #1,5,5: PRINT #1, "voulez vous rejouer (O/N)?";
240  R$=INKEY$: IF R$="" THEN 240
250 IF R$<>"N" THEN 130
260 END
270 '
5000 '------------ SP LECTURE REPONSE ----------------------
5010 '
5020 INK 2,CD: INK 3,CED: PAPER #1,2: PEN #1,3: CLS #1
5030 NL=LEN(LE$)
5040 LOCATE #1, 2,5
5050 WHILE INKEY$<>"": WEND
5060 IF NL=1 THEN PRINT #1, "quelle etait la lettre?";
5070 IF NL>1 THEN PRINT#1,"quelles etaient les";NL;"lettres";
5080 LOCATE #1, 2,8
5090 FOR I=1 TO NL
5100  R$=INKEY$: IF R$="" THEN 5100
5110  PRINT #1, R$;
5120  IF R$<>MID$(LE$,I,1) THEN 5230
5130 NEXT I
5140 '
5150 LOCATE #1, 2,10
5160 PRINT#1, R$(INT(RND(1)*NR+1))
5170 FOR K=1 TO 100: NEXT K
5180 SOUND 1,478,9: SOUND 1,379,9: SOUND 1,319,9:SOUND 1,239,9
5190 OK=1
```

```
5200 FOR K=1 TO 100: NEXT K
5210 RETURN
5220 '
5230 LOCATE #1, 2,10
5240 PRINT #1, N$(INT(RND(1)*NN+1))
5250 FOR H=200 TO 500 STEP 10: SOUND 1,H,2: NEXT H
5260 FOR K=1 TO 200: NEXT K
5270 OK=0
5280 LOCATE #1, 2,11: PRINT #1, "LA BONNE REPONSE ETAIT";
5290 LOCATE #1, 2,13: GOSUB 6500
5300 RETURN
5310 '
6000 '------------ SP PREPARATION AFFICHAGE LETTRES ----------
6010 '
6020 INK 2,CD: INK 3,CED: PAPER #1,2: PEN #1,3: CLS#1
6030 LOCATE #1, 11,8: PRINT #1, "R E G A R D E Z";
6040 FOR K=1 TO 100: NEXT K
6050 '
6060 CE=CE(INT(RND(1)*NE+1))
6070 CF=CF(INT(RND(1)*NF+1))
6080 C=65+INT(RND(1)*26)
6090 LE$=LE$+CHR$(C)
6100 LOCATE 12,4: PRINT LEN(LE$)
6110 INK 2,CE: INK 3,CF: PAPER #1,2: PEN #1,3
6120 CLS #1
6130 LOCATE #1, 3,7
6140 RETURN
6150 '
6500 '------------ SP AFFICHAGE LETTRES ----------------------
6510 '
6520 FOR I=1 TO LEN(LE$)
6530  C$=MID$(LE$,I,1): PRINT #1, C$;
6540  C=ASC(C$)
6550  H=10*(C-50)
6560  SOUND 1,H,10
6570  FOR K=1 TO 150: NEXT K
6580 NEXT I
6590 FOR K=1 TO 1000: NEXT K
6600 RETURN
6610 '
7500 '------------ SP FIN D'UN COUP -------------------------
7510 '
7520 P=LEN(LE$)-1
7530 LOCATE #1,2,15: PRINT #1,"vous marquez"; P; "POINTS";
7540 FOR K=1 TO 100: NEXT K
```

```
7550 '
7560 SC=SC+P
7570 LOCATE 24,2: PRINT SC;
7580 FOR K=1 TO 200: NEXT K
7590 IF CC=NC THEN RETURN
7600 LOCATE #1, 2,18: INK 2,CD: INK 3,CED
7610 PRINT #1, "tapez sur une touche pour continuer";;
7620 R$=INKEY$: IF R$="" THEN 7620
7630 RETURN
7640 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 CLS: CLS #1
8030 LOCATE 5,2: PRINT "COUP";
8040 LOCATE 5,4: PRINT "LETTRES";
8050 LOCATE 17,2: PRINT "SCORE";
8060 LOCATE 17,4: PRINT "M-SCORE";
8070 LOCATE 24,4: PRINT MS;
8080 SC=0
8090 LOCATE 24,2: PRINT "      ";
8100 RETURN
8110 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 CES=7: CS=1: CED=7: CD=1: BR=16
9030 NC=10: MS=0
9040 '
9050 DATA 6,0,1,2,3,4,9
9060 READ NF: DIM CF(NF)
9070 FOR I=1 TO NF
9080  READ CF(I)
9090 NEXT I
9100 '
9110 DATA 13,10,11,12,13,14,15,16,17,20,21,24,25,26
9120 READ NE: DIM CE(NE)
9130 FOR I=1 TO NE
9140  READ CE(I)
9150 NEXT I
9160 '
9170 DATA 9
9180 DATA OK,OUI,EXACT,BIEN,JUSTE,CORRECT,BRAVO,PARFAIT,VRAI
9190 READ NR: DIM R$(NR)
9200 FOR I=1 TO NR
9210  READ R$(I)
9220 NEXT I
```

```
9230 '
9240 DATA 5
9250 DATA NON,FAUX,INEXACT,INCORRECT,ERREUR
9260 READ NM: DIM N$(NM)
9270 FOR I=1 TO NM
9280  READ N$(I)
9290 NEXT I
9300 '
9310 MODE 1: CLS
9320 INK 0,CES: INK 1,CS
9330 PAPER 0: PEN 1: BORDER BR
9340 WINDOW #1, 1,40,6,25
9350 PEN #1, 0: PAPER #1, 1
9360 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

▷ **Pour modifier le temps qui s'écoule entre l'affichage de deux lettres successives.**

Remplacez, en 6570, le nombre 50 par une valeur de votre choix. Par exemple :

```
6570 FOR K=1 TO 100 : NEXT K
```

conduira approximativement à un rythme deux fois plus lent.

De façon analogue :

```
6570 FOR K=1 TO 25 : NEXT K
```

conduira à un rythme deux fois plus rapide.

▷ **Pour modifier le temps d'attente à la fin de la présentation d'une suite de lettres**

Remplacez, en 6590, la valeur 2000 par un nombre de votre choix.

▷ **Pour supprimer le message "REGARDEZ" qui précède chaque séquence de lettres**

Supprimez les lignes 6020 à 6040.

▷ **Pour modifier le nombre d'essais proposés**

Remplacez, en 9030, la valeur 10 (dans NC = 10) par une valeur de votre choix.

▷ **Pour modifier les couleurs employées** lors de la présentation des suites de lettres, sachez que :

- l'instruction DATA de la ligne 9050 contient le nombre de couleurs de fond (ici 6) suivi des codes de ces couleurs (ici 0, 1, 2, 3, 4 et 9).
- l'instruction DATA de la ligne 9110 contient le nombre de couleurs d'écriture (ici 13) suivi des codes de ces couleurs (ici 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 24, 25 et 26).

REMARQUE : évitez de placer deux codes de couleurs identiques dans chacune de ces instructions car, dans certains cas, les lettres proposées seraient invisibles.

▷ **Pour obtenir des suites de chiffres à la place des suites de lettres**

Remplacez la ligne 6080 par :

```
6090 C=48 + INT (RND (1) * 10)
```

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard.

b) Le programme principal (100-260)

120 appelle le sous-programme d'initialisation du jeu.

130-250 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez pas rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

140-200 proposent les dix essais. Pour chaque essai :

150 initialise la suite de lettres proposées.

160 affiche le numéro du coup.

170-180 ajoutent une lettre à la suite (sous-programme 6000), l'affichent (6500) et lisent votre réponse (5000) et ceci jusqu'à ce que votre réponse soit fausse.

190 appelle le sous-programme de fin d'un coup.

210 actualise le meilleur score.

220-240 vous demandent si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9360)

9010 détermine les couleurs.

9030 fixe le nombre de coups et initialise le meilleur score.

9040-9090 déterminent les différentes couleurs de fond qui seront utilisées lors de la présentation des lettres.

9100-9150 déterminent les différentes couleurs d'écriture qui seront utilisées lors de la présentation des lettres.

9160-9220 déterminent les différents messages de ''compliments'' formulés en cas de réponse exacte.

9230-9290 déterminent les différents messages formulés en cas de réponse fausse.

9310-9350 définissent la fenêtre numéro 1 (où se déroulera le jeu) et fixent les couleurs.

d Le sous-programme d'initialisation d'une partie (8000-8100)

8020-8060 effacent l'écran et affichent les différents titres (COUP, LETTRES, SCORE, M-SCORE).

8070-8090 affichent le meilleur score et initialisent le score.

e) Le sous-programme de préparation de la suite de lettres (6000-6140)

6020-6040 affichent le mot ''REGARDEZ'' pendant un court instant.

6060-6070 choisissent une couleur de fond et une couleur d'encre.

6080-6090 tirent au hasard une lettre de l'alphabet et l'ajoutent à la suite déjà proposée.

6100 affiche le nombre de lettres de la suite.

6110-6120 effacent l'écran et en modifient les couleurs.

6130 positionne le curseur.

f) Le sous-programme d'affichage des lettres (6500-6600)

6520-6580 affichent les différentes lettres. Pour chaque lettre :

6530 affiche la lettre concernée.

6530-6570 émettent un son dont la hauteur dépend de la lettre.

6590 attend un instant.

g) Le sous-programme de lecture de la réponse (5000-5310)

5020 modifie les couleurs de l'écran.

5030 détermine la longueur de la suite de lettres.

5040-5070 vous demandent quelles étaient les lettres proposées.

5080-5130 lisent chacune des lettres en s'assurant qu'elles sont correctes.

5150-5210 exécutées en cas de réponse correcte, ces lignes affichent un message de compliments, font retentir un petit accord et placent à 1 l'indicateur OK.

5230-5290 exécutées en cas de réponse incorrecte, ces lignes affichent un message d'erreur, émettant un son approprié, placent à 0 l'indicateur OK et vous affichent (en musique) la bonne réponse.

h) Le sous-programme de fin d'un coup (7500-7630)

7520-7540 vous affichent le nombre de points marqués pour l'essai en cours et attendent un instant.

7560-7580 actualisent votre score et attendent un instant.

7590-7620 attendent que vous frappiez une touche (sauf s'il s'agit du dernier coup de la partie).

5. Liste des variables

CES	Couleur de fond de la partie supérieure où s'affichent numéro du coup, nombre de lettres, score et meilleur score.
CS1	Couleur d'affichage utilisée dans cette partie supérieure.
CED	Couleur de fond utilisée lors de la lecture de votre réponse.
CD	Couleur d'affichage de votre réponse.
BR	Couleur du tour de l'écran.
NC	Nombre de coups d'une partie.
MS	Meilleur score.
NF	Nombre de couleurs de fond utilisées lors de la présentation des lettres.
CF(CF)	Tableau contenant les codes de ces différentes couleurs.

NE	Nombre de couleurs utilisées pour l'affichage des lettres.
CE(NE)	Tableau contenant les codes de ces différentes couleurs.
NR	Nombre de messages de ''compliments''.
R$(NR)	Tableau contenant ces différents messages.
NN	Nombre de messages ''d'erreur''.
N$(NN)	Tableau contenant ces différents messages.
SC	Score.
CC	Numéro de l'essai en cours.
LE$	Chaîne contenant la suite des lettres proposées.
OK	Indicateur utilisé à la fin d'un essai : 0 : la suite de lettres n'a pas été convenablement reconstituée. 1 : la suite de lettres a été convenablement reconstituée.
CE	Couleur de fond utilisée pour afficher les lettres pour l'essai en cours.
CF	Couleur utilisée pour afficher les lettres pour l'essai en cours.
SC	Score.

Variables auxiliaires

I A R$ C C$

NT P

9
Les allumettes suédoises

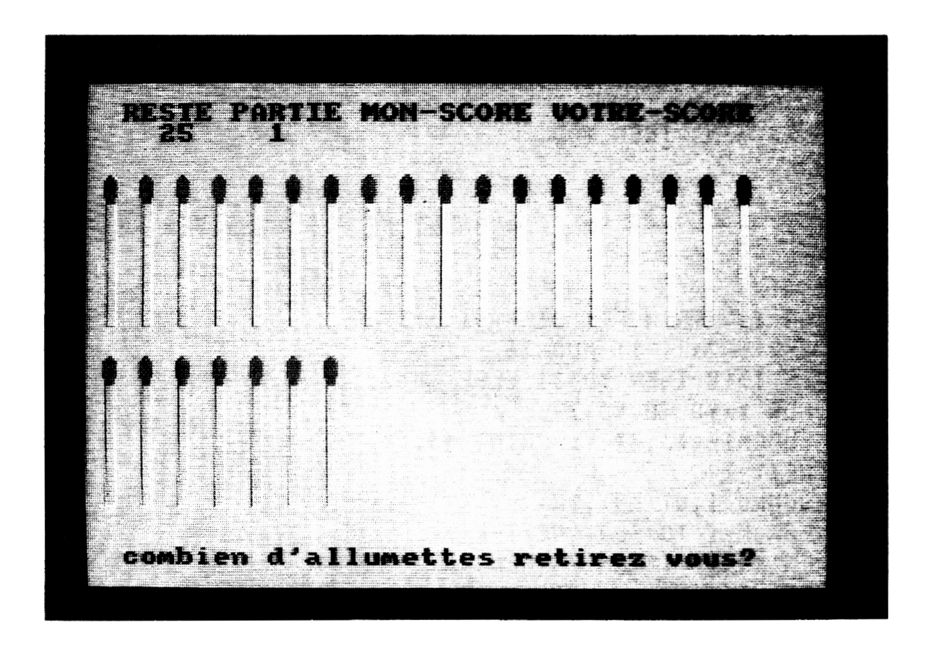
1. Le jeu

Nous vous proposons une version simplifiée du célèbre jeu, connu également sous le nom de ''Marienbad''. Ce programme vous donnera une bonne idée de la qualité des dessins que peut réaliser votre AMSTRAD.

Le jeu est prévu, en principe, pour deux joueurs. Ici, c'est l'ordinateur (ou plutôt le programme) qui tiendra la place de votre adversaire.

Au départ, 36 allumettes sont présentes. Chaque joueur (l'ordinateur et vous) choisit à tour de rôle d'enlever un nombre d'allumettes compris entre 1 et 7. Celui qui se trouve contraint d'enlever la dernière allumette a perdu.

Le programme commence par dessiner les 36 allumettes puis choisit au hasard celui qui jouera en premier. Quand c'est au programme de jouer, il vous dit : ''j'enlève x allumettes''. Vous voyez ensuite les allumettes concernées disparaî-



tre en musique (chaque allumette égrène une note de la gamme). Quand c'est à votre tour de jouer, vous voyez apparaître la question ''combien d'allumettes retirez-vous ?''. Vous répondez simplement en tapant sur le chiffre correspondant. Notez que toute touche ne correspondant pas à un nombre compris entre 1 et 7 fait apparaître le message ''svp entre 1 et 7''.

Le programme affiche en permanence le nombre d'allumettes restant en jeu (RESTE), le numéro de la partie en cours ainsi que le nombre de parties gagnées par chacun des partenaires.

2. Le programme

```
100 '****************** ALLUMETTES SUEDOISES *****************
110 '
120 GOSUB 9000
130  GOSUB 8000
140   IF JO=1 THEN GOSUB 5000 ELSE GOSUB 4000
150   GOSUB 3000
160   N=N-P
170   LOCATE #1, 4,3
180   PRINT #1, N; " ";
190   JO=-JO
```

```
200  IF N>1 THEN 140
210  GOSUB 8500
220  JD=-JD
230 IF R$<>"N" THEN 130
240 END
250 '
3000 '------------ SP EFFACEMENT DE P ALLUMETTES -------------
3010 '
3020 KN=1
3030 FOR I=N TO N-P+1 STEP -1
3040  X=2 + 2*(I-1) MOD 36
3050  IF I>18 THEN YD=11 ELSE YD=2
3060  FOR Y=YD TO YD+7
3070   LOCATE #3, X,Y: PRINT #3, " ";
3080  NEXT Y
3090  SOUND 1,NT(KN),10
3100  FOR K=1 TO 300: NEXT K
3110  KN=KN+1: IF KN>7 THEN KN=1
3120 NEXT I
3130 RETURN
3140 '
4000 '------------ SP LECTURE CHOIX DU JOUEUR ----------------
4010 '
4020 CLS #2: LOCATE #2, 3,2
4030 PRINT #2, "combien d'allumettes retirez vous?";
4040 R$=INKEY$: IF R$="" THEN 4040
4050 P=VAL(R$)
4060 IF P>0 AND P<=PM THEN 4100
4070 LOCATE #2,15,3
4080 PRINT #2, "entre 1 et"; PM;
4090 GOTO 4040
4100 LOCATE #2, 38,2: PRINT #2, R$;
4110 FOR K=1 TO 1000: NEXT K
4120 CLS #2
4130 RETURN
4140 '
5000 '------------ SP CHOIX ORDINATEUR DE P ALLUMETTES -------
5010 '
5020 Q=INT(N/(PM+1))
5030 R=N-Q*(PM+1)
5040 P=R-1: IF P<=0 THEN P=INT(RND(1)*PM+1)
5050 IF N<=PM+1 THEN P=N-1
5060 '
5070 FOR I=1 TO RND(1)*400: NEXT I
5080 CLS#2: LOCATE #2, 3,2
```

```
5090 PRINT #2, "j'enleve"; P; "allumette";
5100 IF P>1 THEN PRINT #2, "s";
5110 FOR K=1 TO 800: NEXT K
5120 RETURN
5130 '
7000 '------------ SP DESSIN DE NX ALLUMETTES ----------------
7010 '
7020 X=2: YD=2
7030 FOR I=1 TO NX
7040  PEN #3, 2
7050  LOCATE #3, X,YD: PRINT #3, TH$;
7060  LOCATE #3, X,YD+1: PRINT #3, TB$;
7070  PEN #3, 1
7080  FOR Y=YD+2 TO YD+7
7090   LOCATE #3, X,Y: PRINT #3, B$;
7100  NEXT Y
7110  X=X+2
7120  IF I=18 THEN X=2: YD=11
7130 NEXT I
7140 RETURN
7150 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 CLS #2: CLS #3
8030 N=NX: NP=NP+1: KN=1
8040 LOCATE #1, 4,3: PRINT #1, N;
8050 LOCATE #1, 10,3: PRINT #1, NP;
8060 GOSUB 7000
8070 '
8080 LOCATE #2, 3,1
8090 IF JD=1 THEN PRINT #2, "je joue en premier";
8100 IF JD=0 THEN PRINT #2, "vous jouez en premier";
8110 JO=JD
8120 FOR K=1 TO 800: NEXT K
8130 LOCATE #2, 3,3
8140 CLS #2
8150 RETURN
8160 '
8500 '------------ SP FIN DE PARTIE -------------------------
8510 '
8520 FOR K=1 TO 1000: NEXT K
8530 FOR Y=2 TO 9
8540  LOCATE #3, 2,Y: PRINT #3, " ";
8550 NEXT Y
8560 '
```

```
8570 IF N=1 AND JO=1 THEN PJ=PJ+1 ELSE PO=PO+1
8580 LOCATE #1, 19,3: PRINT #1, PO;
8590 LOCATE #1, 31,3: PRINT #1, PJ;
8600 '
8610 CLS #2: LOCATE #2, 4,2
8620 PRINT #2, "voulez vous rejouer";
8630 R$=INKEY$: IF R$="" THEN 8630
8640 RETURN
8650 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 BR=0: CE=5: CS=1: CT=6: CB=24
9030 NX=36: PM=7: NP=0: PO=0: PJ=0
9040 '
9050 MODE 1
9060 INK 0,CE: INK 1,CB: INK 2,CT: INK 3,CS
9070 WINDOW #1, 1,40,1,3
9080 WINDOW #2, 1,40,23,25
9090 WINDOW #3, 1,40,4,22
9100 PAPER #1,0: PEN #1,3: CLS #1
9110 PAPER #2,0: PEN #2,3: CLS #2
9120 PAPER #3,0: CLS#3
9130 '
9140 IF RND(1)>0.5 THEN JD=1 ELSE JD=-1
9150 LOCATE #1,3,2
9160 PRINT #1,"RESTE PARTIE MON-SCORE VOTRE-SCORE";
9170 '
9180 SYMBOL AFTER 160
9190 SYMBOL 160, 0,0,0,0,48,120,252,252
9200 SYMBOL 161, 252,252,252,252,252,120,120,120
9210 SYMBOL 162, 120,120,120,120,120,120,120,120
9220 TH$=CHR$(160): TB$=CHR$(161): B$=CHR$(162)
9230 '
9240 DIM NT(7)
9250 DATA 478,426,379,358,319,283,253
9260 FOR I=1 TO 7: READ NT(I): NEXT I
9270 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur de la zone où apparaissent les allumettes**

Remplacez, en 9020, la valeur 5 (dans CE = 5) par un nombre de votre choix compris entre 0 et 26.

▷ **Pour modifier le nombre total d'allumettes**

Remplacez, en 9030, le nombre 36 (dans NX = 36) par une valeur de votre choix ne dépassant pas 36.

▷ **Pour modifier la prise maximum autorisée** (nombre maximum d'allumettes que l'on peut prendre en une seule fois)

Remplacez, en 9030, le nombre 7 (dans PM = 7) par une valeur de votre choix ne dépassant pas 9.

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard.

b) Le programme principal (100-250)

120 appelle le sous-programme d'initialisation du jeu.

130-230 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie

140-200 répètent le ''jeu d'un coup'' (par l'ordinateur ou par vous) jusqu'à ce qu'il ne reste plus qu'une allumette :

140 permet le choix du nombre d'allumettes à retirer, soit par vous (sous-programme 4000), soit par l'ordinateur (sous-programme 5000). Le joueur est précisé par la variable JO (1 : ordinateur, - 1 : vous).

150 appelle le sous-programme qui efface le nombre d'allumettes voulu.

160-180 actualisent et affichent le nombre d'allumettes restantes.

190 change de joueur.

200 examine si la partie est terminée.

210 appelle le sous-programme de fin de partie.

220 change le joueur qui doit débuter.

230 examine si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9270)

9020 détermine les couleurs.

9030 fixe le nombre total d'allumettes, la prise maximum autorisée et initialise les compteurs de parties gagnées.

9050-9120 déterminent les emplacements des 3 fenêtres et en fixent les couleurs.

9140 choisit, au hasard, le joueur qui commencera la première partie.

9140-9150 affichent les différents titres.

9180-9220 fabriquent les caractères graphiques utilisés pour représenter une allumette.

9240-9260 placent, dans le tableau NT, les sept notes de la gamme.

d) Le sous-programme d'initialisation d'une partie (8000-8150)

8020 efface l'écran.

8030-8050 initialisent le nombre d'allumettes, actualisent le numéro de partie et affichent ces deux valeurs.

8060 appelle le sous-programme de dessin de l'ensemble des allumettes.

8080-8110 précisent qui doit jouer le premier.

8120-8140 attendent un instant et effacent l'écran.

e) Le sous-programme de dessin des allumettes (7000-7140)

La variable NX contient le nombre d'allumettes à dessiner. Si ce nombre ne dépasse pas 18, les allumettes sont présentées sur une seule rangée ; dans le cas contraire, elles sont présentées sur deux rangées.

7020 initialise les coordonnées d'une allumette.

7030-7130 dessinent toutes les allumettes. Pour chacune d'entre elles :

7040-7060 dessinent la tête.

7070-7100 dessinent la tige.

7110 incrémente l'abscisse d'une allumette.

7120 tient compte d'un éventuel changement de rangée.

f) Le sous-programme de choix de l'ordinateur (5000-5120)

5020-5050 déterminent le nombre d'allumettes à retirer.

5060-5110 affichent ce nombre et attendent un instant.

g) Le sous-programme de choix du joueur (4000-4130)

4020-4050 vous demandent combien d'allumettes vous voulez enlever.

4060-4090 s'assurent que votre réponse est correcte, en vous précisant s'il y a lieu, les valeurs autorisées.

4100-4120 affichent votre réponse puis, après un court instant, effacent l'écran.

h) Le sous-programme d'effacement de "P" allumettes (3000-3130)

3020 initialise le numéro de la note à jouer (en accompagnement de l'effacement d'une allumette).

3030-3120 effacent les différentes allumettes. Pour chaque allumette :

3040-3050 déterminent les coordonnées de l'allumette à effacer.

3060-3080 effacent cette allumette.

3090 joue la note correspondante.

3100 attend un instant.

3110 incrémente le numéro de note.

i) Le sous-programme de fin de partie (8500-8640)

8520-8560 efface la dernière allumette, après un court instant.

8570-8600 actualisent le nombre de parties gagnées (par l'ordinateur ou par vous) et en affichent les valeurs.

8610-8630 vous demandent si vous voulez rejouer.

5. *Liste des variables*

BR	Couleur du tour de l'écran.
CE	Couleur de fond des 3 fenêtres.
CS	Couleur d'affichage des scores et des messages.
CT	Couleur de la "tête" des allumettes.
CB	Couleur de la tige des allumettes.
NX	Nombre total d'allumettes.

PM	Prise maximum autorisée.
NP	Nombre de parties jouées.
PO	Nombre de parties gagnées par l'ordinateur.
PJ	Nombre de parties gagnées par le joueur.
JD	Joueur qui doit jouer en premier dans la prochaine partie : 1 : ordinateur — 1 : vous
TH$	Caractère graphique représentant la partie supérieure de la tête d'une allumette.
TB$	Caractère graphique représentant la partie inférieure de la tête d'une allumette.
B$	Caractère graphique servant à représenter la tige d'une allumette.
NT(7)	Tableau contenant les hauteurs des sept notes de la gamme.
N	Nombre d'allumettes restantes.
JO	Joueur ''courant'' (1 : ordinateur, — 1 : vous).
P	Nombre d'allumettes à enlever.
X	Abscisse d'une allumette.
YD	Ordonnée courante d'un emplacement écran.

Variables auxiliaires

R$ A I

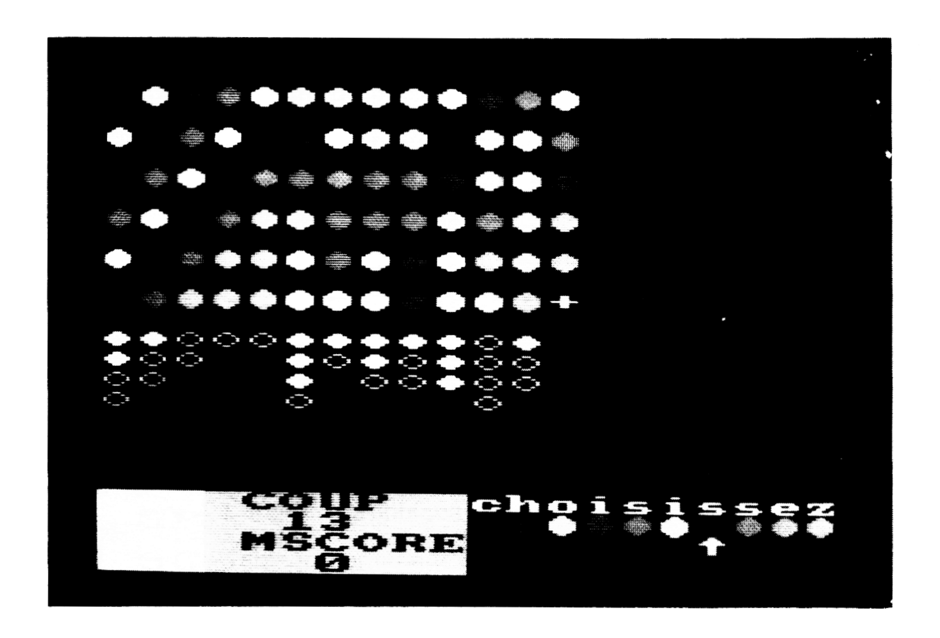
Q R Y

10
MASTER MIND

1. Le jeu

Il s'agit là d'un grand classique des jeux de réflexion. Si vous en connaissez déjà les règles, vous retrouverez un univers familier. Par rapport au jeu manuel vous rencontrerez essentiellement deux différences :

- Le nombre de positions peut atteindre 6 tandis que le nombre de couleurs peut aller jusqu'à 9.
- Manuellement, vous preniez un pion pour le déposer dans un trou du plateau de jeu. Ici, vous sélectionnez la couleur de votre pion grâce à un curseur que vous déplacez à l'aide des touches → et ←. Cette façon de procéder évite la sempiternelle question ''entrez les couleurs choisies'' qui oblige toujours à coder les couleurs (chiffres, lettres...).



Le but du jeu

Il consiste à découvrir une combinaison de couleurs, tirée en secret par l'ordinateur. Pour cela, vous formulez des propositions de combinaison. A chaque fois, vous êtes renseigné sur :

— le nombre de couleurs exactes situées en bonne position,

— le nombre de couleurs exactes, mais situées en mauvaise position.

Le déroulement d'une partie

A chaque partie, le programme vous laisse déterminer la difficulté du jeu en vous demandant le nombre de positions et le nombre de couleurs que vous souhaitez utiliser. Ce choix vous sera rappelé, tout au long de la partie, en bas et à gauche de l'écran. Ainsi, par exemple, 5P signifiera 5 positions et 4C correspondra à 4 couleurs. D'autre part, la ''palette'' des couleurs avec lesquelles vous pouvez jouer (compte tenu de votre choix) apparaît en bas à droite de l'écran.

Le programme choisit ensuite une combinaison de couleurs (les répétitions sont possibles). Pour deviner cette combinaison, vous effectuez une proposition en choisissant parmi les couleurs de la palette. Pour cela, vous amenez la ''flèche noire'' sous la couleur voulue à l'aide des touches →et ← et vous la sélectionnez en appuyant sur la barre d'espace. Un pion de la couleur choisie apparaît alors en haut à gauche, à l'emplacement qui vous était signalé par une croix.

Cette croix apparaît maintenant un peu plus bas pour vous montrer que le programme attend que vous choisissiez une seconde couleur. Cela se poursuit jusqu'à ce que vous ayez autant de couleurs qu'il y a de positions prévues.

L'ordinateur vous fournit alors sa réponse sous forme de petits pions accompagnés de musique. Le rythme d'apparition de ces pions est irrégulier, ce qui donne l'impression que l'ordinateur ''réfléchit'' avant de répondre. La signification des pions est la suivante :

- une couleur à sa place pour chaque petit pion blanc ;
- une bonne couleur, non à sa place, pour chaque petit pion noir cerclé de blanc.

Si, par malchance (ou peut-être par chance !), votre proposition ne comporte aucune bonne couleur, vous entendrez simplement ''zap''.

Le programme vous demandera ensuite de formuler une autre proposition (à moins que vous ne soyez tombé juste au premier coup !). Le Jeu se poursuivra ainsi jusqu'à ce que vous trouviez la bonne combinaison ou que vous dépassiez le nombre maximum (20) de propositions. Dans ce dernier cas, le programme vous fournira sa combinaison cachée et vous pourrez observer tout à loisir l'écran rempli de pions de couleurs et éventuellement analyser votre jeu (peut-être, même, vérifier que le programme n'a pas menti ! eh ! on ne sait jamais...).

Remarque : en cas d'erreur

Si vous vous trompez (ou si vous changez d'avis !) pendant que vous formulez votre proposition, vous pouvez toujours l'annuler ; il vous suffit pour cela de taper sur ''ESC''.

2. Le programme

```
100 '********************** MASTER MIND ***********************
110 '
120 GOSUB 9000
130  GOSUB 8000
140  CP=1
150 '
160   GOSUB 5000: GOSUB 4000
170   CP=CP+1
180  IF PX<>NP AND CP<=NX THEN 160
190 '
200  IF PX=NP THEN GOSUB 2000 ELSE GOSUB 3000
210  FOR K=1 TO 2000: NEXT K
220  LOCATE #3,1,1
```

```
230  PRINT #3,"ON REJOUE";
240  R$=INKEY$: IF R$="" THEN 240
250 IF R$<>"N" THEN 130
260 END
270 '
2000 '------------ SP GAGNE -----------------------------------
2010 '
2020 FOR I=1 TO 50
2030  H=20+500*RND(1): SOUND 1,H,5
2040 NEXT I
2050 LOCATE 11,24: PRINT "GAGNE en";
2060 LOCATE 11,25: PRINT CP-1;"coups";
2070 IF MS(NP,NC)>CP-1 OR MS(NP,NC)=0 THEN MS(NP,NC)=CP-1
2080 RETURN
2090 '
3000 '------------ SP PERDU -----------------------------------
3010 '
3020 FOR H=10 TO 400 STEP 1: SOUND 1,H,1: NEXT H
3030 LOCATE 11,25: PRINT "PERDU";
3040 LOCATE #3,1,2: PRINT #3,"C'etait : ";
3050 LOCATE 11,23: PRINT "          ";
3060 FOR IC=1 TO NP
3070  LOCATE 11+IC,23
3080  PEN TI(IC)-1: PRINT P$;
3090 NEXT IC
3100 RETURN
3110 '
4000 '------------ SP ANALYSE COMBINAISON --------------------
4010 '
4020 FOR I=1 TO NP: TP(I)=TI(I): NEXT I
4030 '
4040 PX=0
4050 FOR I=1 TO NP
4060  IF TP(I)=PR(I) THEN PX=PX+1: TP(I)=0: PR(I)=-1
4070 NEXT I
4080 '
4090 CX=0
4100 FOR I=1 TO NP
4110  FOR J=1 TO NP
4120   IF PR(I)=TP(J) THEN CX=CX+1: TP(J)=0: GOTO 4140
4130  NEXT J
4140 NEXT I
4150 '
4160 YR=2*(NP+1)
```

```
4170 IF PX=0 THEN 4250
4180  FOR I=1 TO PX
4190   FOR K=1 TO TM*RND(1): NEXT K
4200   PEN 10: LOCATE CP,YR: PRINT PP$;
4210   SOUND 1,478,10
4220   YR=YR+1
4230  NEXT I
4240 '
4250 IF CX=0 THEN 4330
4260  FOR I=1 TO CX
4270   FOR K=1 TO TM*RND(1): NEXT K
4280   PEN 10: LOCATE CP,YR: PRINT PC$;
4290   SOUND 1,379,10
4300   YR=YR+1
4310  NEXT I
4320 '
4330 IF PX+CX<>0 THEN 4370
4340  FOR H=500 TO 200 STEP -10
4350   SOUND 1,H,1
4360  NEXT H
4370 RETURN
4380 '
5000 '------------ SP LECTURE COMBINAISON PROPOSEE -----------
5010 '
5020 LOCATE #3,1,2
5030 PRINT #3,"choisissez";
5040 LOCATE #2,2,2: PRINT #2,CP;
5050 '
5060 IP=1
5070  LOCATE CP,2*IP
5080  PEN 10: PRINT "+";
5090  GOSUB 7000
5100  IF ANUL=1 THEN 5160
5110   PR(IP)=CL
5120   LOCATE CP,2*IP
5130   PEN CL-1: PRINT P$;
5140  GOTO 5210
5150 '
5160  FOR K=1 TO IP
5170   LOCATE CP,2*K: PRINT " ";
5180  NEXT K
5190  IP=0
5200 '
5210  IP=IP+1: IF IP<=NP THEN 5070
5220 RETURN
```

```
5230 '
7000 '------------ SP ENTREE D'UNE COULEUR -------------------
7010 '
7020 ANUL=0
7030  PEN 10: LOCATE 11+CL,24
7040  PRINT CHR$(94);
7050  R$=INKEY$: IF R$=""THEN 7050
7060  R=ASC(R$)
7070  LOCATE 11+CL,24: PRINT " ";
7080  CL=CL-(R=242)*(CL>1)+(R=243)*(CL<NC)
7090 IF R<>32 AND R<>127 THEN 7030
7100 IF R=127 THEN ANUL=1
7110 RETURN
7120 '
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 PAPER 9: CLS
8030 T$="MASTERMIND"
8040 LOCATE 6,8
8050 FOR I=1 TO LEN(T$)
8060  PEN I MOD 9: PRINT MID$(T$,I,1);
8070 NEXT I
8080 '
8090 LOCATE 2,14: PRINT "combien de";
8100 LOCATE 2,16: PRINT "positions (2-6)";
8110 R$=INKEY$: IF R$="" THEN 8110
8120 NP=VAL(R$): IF NP<2 OR NP>6 THEN 8110
8130 PRINT NP;
8140 '
8150 LOCATE 2,19: PRINT "combien de";
8160 LOCATE 2,21: PRINT "couleurs (2-9)";
8170 R$=INKEY$: IF R$="" THEN 8170
8180 NC=VAL(R$): IF NC<2 OR NC>9 THEN 8170
8190 PRINT NC;
8200 '
8210 PAPER 9:CLS
8220 WINDOW #1,1,3,22,25: PAPER #1,11: PEN #1,12: CLS #1
8230 WINDOW #2,4,10,22,25: PAPER #2,13: PEN #2,14: CLS #2
8240 WINDOW #3,11,20,21,22: PEN #3,15: PAPER #3,9: CLS#3
8250 '
8260 LOCATE #1,1,2: PRINT #1,NP;
8270 LOCATE #1,3,2: PRINT #1,"P";
8280 LOCATE #1,1,4: PRINT #1,NC;
8290 LOCATE #1,3,4: PRINT #1,"C";
```

```
8300 LOCATE #2,2,1: PRINT #2,"COUP";
8310 LOCATE #2,2,3: PRINT #2,"MSCORE";
8320 LOCATE #2,3,4: PRINT #2,MS(NP,NC);
8330 '
8340 FOR I=1 TO NC
8350  LOCATE 11+I,23
8360  PEN I-1: PRINT P$;
8370 NEXT I
8380 '
8390 FOR I=1 TO NP
8400  TI(I)=INT(RND(1)*NC+1)
8410 NEXT I
8420 RETURN
8430 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 MODE 0
9030 CE=0: BR=0: CE1=14: CS1=2: CE2=16: CS2=1: CD=23: CR=25
9040 '
9050 DIM TI(6), PR(6), TP(6), MS(6,9)
9060 CL=1: NX=20: TM=800
9070 '
9080 SYMBOL AFTER 160
9090 SYMBOL 160, 24,60,126,126,126,126,60,24
9100 SYMBOL 161, 0,24,36,66,66,36,24,0
9110 SYMBOL 162, 0,24,60,126,126,60,24,0
9120 P$=CHR$(160): PC$=CHR$(161): PP$=CHR$(162)
9130 '
9140 DATA 4,5,6,7,8,9,15,16,21
9150 FOR I=0 TO 8
9160  READ CO: INK I,CO
9170 NEXT I
9180 INK 9,CE: INK 10,CR: INK 11,CE1: INK 12,CS1
9190 INK 13,CE2: INK 14,CS2: INK 15,CD
9200 BORDER BR
9210 RETURN
```

3. Si vous souhaitez personnaliser ce programme

◁ **Pour modifier la couleur des pions**

Remplacez les 9 nombres de l'instruction 9140 par neuf valeurs de votre choix, en veillant à ce qu'elles soient toutes différentes.

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard.

b) Le programme principal (100-260)

120 appelle le sous-programme d'initialisation du jeu.

130-250 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

140 initialise le numéro du coup et les coordonnées d'affichage d'une combinaison.

160-180 sont répétées jusqu'à ce que vous ayez gagné ou que le nombre de coups joués atteigne la limite 20 :

160 appelle les sous-programmes de lecture et d'analyse de la combinaison proposée.

170 incrémente le compteur de coups.

200 appelle suivant le cas, l'un des sous-programmes ''gagné'' ou ''perdu''.

210-240 vous demandent si vous voulez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9210)

9030 détermine les différentes couleurs.

9050-9060 réservent les différents tableaux, fixent la position initiale de la flèche, le nombre maximum de coups, le temps d'attente maximum entre l'affichage de deux petits pions.

9080-9120 fabriquent les caractères graphiques (pion, petit pion plein, petit pion cerclé).

9140-9200 fixent les différentes couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8430)

8020-8070 affichent le titre ''MASTERMIND'' en utilisant les neuf couleurs de la palette.

8090-8130 vous font choisir le nombre de positions.

8140-8190 vous font choisir le nombre de couleurs.

8200-8240 effacent l'écran et fixent les positions des 3 ''fenêtres'' et leurs couleurs.

8250-8290 rappellent, en bas à gauche, le nombre de positions et le nombre de couleurs.

8300-8320 affichent les différents ''titres'' et le meilleur score.

8330-8370 affichent la ''palette'' des couleurs.

8380-8410 tirent au hasard la combinaison à deviner. Notez que chaque couleur de la palette est représentée par un entier compris entre 1 et 9.

e) Le sous-programme de lecture de la combinaison proposée (5000-5220)

5020-5040 affichent le message ''faites votre choix'' et le numéro du coup.

5060 initialise le numéro de position d'un pion, dans la proposition.

5070-5210 sont répétées jusqu'à ce que toutes les positions aient reçu une couleur :

5070-5080 affichent une croix à l'emplacement du prochain pion.

5090 appelle le sous-programme de choix d'une couleur dans la palette proposée.

5100 examine si une demande d'annulation a été formulée.

5110-5130 mémorisent la couleur proposée dans le tableau PR et affichent le pion correspondant.

5160-5190 exécutées en cas d'annulation, ces instructions effacent les pions déjà affichés (pour ce coup) et remettent à zéro le numéro de position.

5210 passe, s'il y a lieu, à la position suivante.

f) Le sous-programme de choix d'une couleur (7000-7110)

7020 initialise l'indicateur d'annulation.

7030-7090 correspondent au déplacement de la flèche de sélection. Elles sont répétées jusqu'à ce que vous frappiez, soit la barre d'espace, soit la touche ''ESC'' :

7030-7040 affichent la flèche dans sa position courante.

7050 attend que vous frappiez une touche.

7060-7080 déterminent la nouvelle position de la flèche.

7090 examine si la touche frappée est ''espace'' ou ''ESC''.

7100 examine si une annulation a été demandée.

g) Le sous-programme d'analyse de la combinaison proposée (4000-4370)

4020 recopie, dans le tableau de travail TP, la combinaison tirée par l'ordinateur. Cela permet d'effectuer certaines modifications dans TP (lors de l'analyse de la proposition) sans altérer le tableau TI.

4040-4070 comptent le nombre (PX) de couleurs bien placées.

4090-4140 comptent le nombre (CX) de couleurs présentes dans la combinaison cachée, mais mal placées. Notez bien qu'il a fallu éviter de prendre en considération les couleurs déjà comptées dans PX.

4160-4230 affichent en musique, s'il y a lieu, un nombre de petits pions blancs égal au nombre de couleurs à leur place.

4250-4310 affichent en musique, s'il y a lieu, un nombre de pions cerclés égal au nombre de bonnes couleurs mal placées.

4330-4360 émettent un ''zap'' si votre proposition ne comporte aucune bonne couleur.

h) Le sous-programme gagné (2000-2080)

2020-2040 émettent quelques sons aléatoires.

2050-2060 affichent ''GAGNÉ'' et le nombre de coups.

2070 actualise le meilleur score s'il y a lieu.

i) Le sous-programme perdu (3000-3100)

3020 émet un son approprié.

3030 affiche ''PERDU''.

3040-3090 affichent la ''bonne'' combinaison.

5. *Liste des variables*

CE	Couleur de fond de la partie de l'écran où s'affichent vos propositions.
BR	Couleur du tour de l'écran.
CE1	Couleur de fond de la partie gauche inférieure où sont récapitulés le nombre de positions et le nombre de couleurs (fenêtre 1).

CE2	Couleur de fond de la partie inférieure où sont affichés le numéro du coup et le meilleur score (fenêtre 2).
CS1	Couleur utilisée pour l'affichage dans la fenêtre 1.
CS2	Couleur utilisée pour l'affichage dans la fenêtre 2.
CD	Couleur utilisée pour l'affichage des ''messages'' (en fenêtre 3).
CR	Couleurs des ''petits pions''.
NX	Nombre maximum de coups.
TM	Temps d'attente maximum utilisé lors de la réponse du programme à une proposition.
TI(6)	Tableau contenant la combinaison tirée par le programme.
PR(6)	Tableau contenant la proposition du joueur.
TP(6)	Tableau ''de travail'' employé dans l'analyse de la proposition.
MS(6,9)	Tableau des meilleurs scores.
CL	Position de la flèche de sélection des couleurs (de 1 à 9).
P$	Caractère graphique représentant un pion.
PP$	Caractère graphique représentant un petit pion plein (''bonne couleur'').
PC$	Caractère graphique représentant un petit pion cerclé (''bonne position'').
NP	Nombre de positions choisies pour la partie en cours.
NC	Nombre de couleurs choisies pour la partie en cours.
CP	Numéro du coup qui est en train de se jouer.
IP	Position d'un pion dans la proposition du joueur.
ANUL	Indicateur d'annulation de la proposition en cours : 0 : pas d'annulation 1 : annulation demandée.
CX	Nombre de bonnes couleurs, mal placées.
PX	Nombre de bonnes couleurs à leur place.
YR	Ordonnée d'un petit pion.

Variables auxiliaires

I T$ R$ A R K

11
Gobe mouches

1. Le jeu

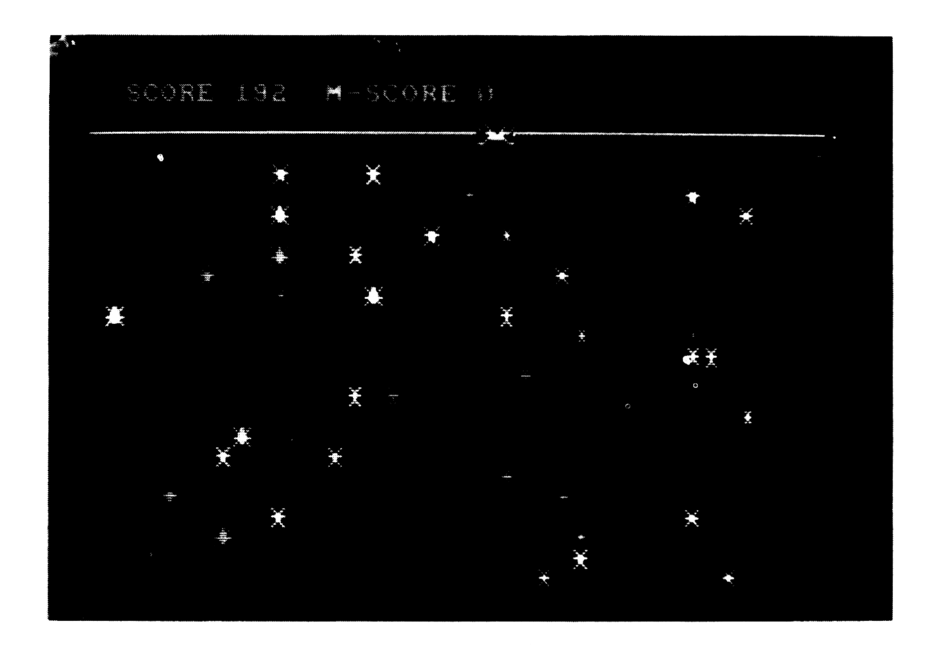
Imaginez que vous soyez une petite araignée obligée de chasser les mouches pour se nourrir.

Heureusement celles-ci pullulent dans votre environnement et vous n'avez que l'embarras du choix. Toutefois vous ne pouvez vous déplacez que le long d'un fil (de votre toile).

Vous vous déplacez le long du haut de l'écran grâce aux touches fléchées :

''←'' et ''→'' vous font avancer à vitesse lente dans la direction indiquée.

'' ↑ '' et '' ↓ '' vous font avancer, dans les mêmes directions, à une vitesse triple de la précédente.



Pour avaler une mouche, il vous suffit de l'intercepter dans son vol (qui se déroule toujours verticalement). Chaque mouche rapporte un nombre de points qui dépend de sa taille et de sa couleur. Les mouches roses procurent de 2 points (pour les plus petites) à 8 points (pour les plus grosses). Les mouches jaunes en procurent deux fois plus (de 4 à 16) et les bleues trois fois plus (de 6 à 24 !).

La partie s'achève au bout d'un temps déterminé à l'avance. Bonne chance et... bon appétit.

2. Le programme

```
100 '**************** GOBE MOUCHES ***************************
110 '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+9,16*(25-Y)+11)
130 GOSUB 9000
140  GOSUB 8000
150  FOR T=1 TO TM
160   IF FNSC(X,4)<>0 THEN XM=X: GOSUB 2000
170   IF FNSC(X+1,4)<>0 THEN XM=X+1: GOSUB 2000
180   XN=X-(INKEY(8)=0)*(X>2)+(INKEY(1)=0)*(X<39)
190   XN=XN-3*((INKEY(0)=0)*(X>4)-(INKEY(2)=0)*(X<37))
```

```
200   IF XN<>X THEN LOCATE #2,X,2: PRINT #2,F$;
210   X=XN
220   LOCATE #2,X,2: PRINT #2,A$;
230   X1=2+RND(1)*38: X2=2+RND(1)*38
240   N1=1+INT(RND(1)*8): N2=1+INT(RND(1)*8)
250   PN1=1+INT(RND(1)*3): PN2=1+INT(RND(1)*3)
260   PEN #3,PN1: LOCATE #3,X1,22: PRINT #3,CHR$(10); M$(N1);
270   PEN #3,PN2: LOCATE #3,X2,22: PRINT #3,M$(N2);
280  NEXT T
290  GOSUB 7000
300 IF R$<>"N" THEN 140
310 END
320 '
2000 '------------ SP MOUCHE GOBEE --------------------------
2010 '
2020 FOR H=150 TO 10 STEP -5: SOUND 1,H,1: NEXT H
2030 XG=16*(XM-1): YG=16*21
2040 K=0
2050 FOR I=3 TO 5: FOR J=4 TO 6
2060  K=K+TEST(XG+2*I,YG+2*J)
2070 NEXT J: NEXT I
2080 SC=SC+K
2090 LOCATE #1,8,1: PRINT #1,SC;
2100 RETURN
2110 '
7000 '------------ SP FIN DE PARTIE -------------------------
7010 '
7020 SOUND 1,478,10: SOUND 1,379,10
7030 SOUND 1,319,10: SOUND 1,239,10
7040 IF SC>MS THEN MS=SC
7050 FOR K=1 TO 1000: NEXT K
7060 CLS #3
7070 LOCATE #3,5,15
7080 WHILE INKEY$<>"": WEND
7090 PRINT #3,"voulez vous rejouer";
7100 R$=INKEY$: IF R$="" THEN 7100
7110 RETURN
7120 '
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 CLS #1: CLS #2: CLS #3
8030 LOCATE #1,3,1: PRINT #1,"SCORE      M-SCORE"; MS
8040 FOR X=1 TO 39 STEP 2
8050  LOCATE #2,X,2: PRINT #2,F$;
8060 NEXT X
```

```
8070 X=19
8080 LOCATE #2, X,2: PRINT #2,A$;
8090 SC=0
8100 RETURN
8110 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 CF=0: BR=0: CA=24: CS=16: C3=20
9030 TM=300: MS=0
9040 '
9050 SYMBOL AFTER 160
9060 SYMBOL 161, 0,0,8,8,0,0,0,0
9070 SYMBOL 162, 0,20,8,28,8,20,0,0
9080 SYMBOL 163, 34,20,8,62,8,20,34,0
9090 SYMBOL 164, 34,20,8,62,8,8,20,34
9100 SYMBOL 165, 66,36,24,126,24,36,66,0
9110 SYMBOL 166, 66,36,24,126,24,24,36,66
9120 SYMBOL 167, 65,34,28,127,28,28,36,65
9130 SYMBOL 168, 153,90,60,60,255,60,90,129
9140 DIM M$(8)
9150 FOR I=1 TO 8: M$(I)=CHR$(160+I): NEXT I
9160 '
9170 SYMBOL 170, 16,136,68,31,7,15,16,225
9180 SYMBOL 171, 8,17,34,125,225,240,8,7
9190 SYMBOL 172, 0,0,0,255,0,0,0,0
9200 A$=CHR$(170)+CHR$(171)
9210 F$=CHR$(172)+CHR$(172)
9220 '
9230 MODE 1: CLS
9240 INK 0,CF: INK 1,CS: INK 2,CA: INK 3,C3
9250 WINDOW #1,1,40,1,1: PAPER #1,0: PEN #1,1
9260 WINDOW #2,1,40,2,3: PAPER #2,0: PEN #2,2
9270 WINDOW #3,1,40,4,25: PAPER #3,0
9280 BORDER BR
9290 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur de l'araignée et de son fil**

Remplacez, en 9020, la valeur 24 (dans CA = 24) par un nombre de votre choix.

▷ **Pour modifier la durée d'une partie**

Remplacez, en 9030, la valeur 300 (dans TM = 300) par un nombre de votre choix.

▷ **Pour que toutes les mouches rapportent le même nombre de points**

Remplacez la ligne 2040 par :

```
2040 K=1
```

et supprimez les lignes 2050 à 2070

▷ **Pour que toutes les mouches soient de la même couleur**

Remplacez la ligne 250 par :

```
250 PN1=... : PN2=...
```

où ... désigne un nombre de votre choix compris entre 1 et 3. Par exemple :

```
260 C1=3 : C2=3
```

produira des mouches bleues.

REMARQUE : notez qu'alors, toutes les mouches rapporteront un nombre de points ne dépendant que de leur taille.

4. *Description du programme*

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Lecture rapide du clavier,
- Examen du contenu de l'écran.

b) Le programme principal (100-320)

120 définit une fonction permettant de connaître la couleur (numéro d'encrier) d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-300 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150 et 280 répètent 300 fois les instructions 160 à 280 :

160-170 examinent si une mouche se trouve en contact avec l'araignée et, le cas échéant, appellent le sous-programme ''mouche gobée'' (2000).

180-190 déterminent la nouvelle position de l'araignée.

200-220 déplacent l'araignée sur son fil.

230-270 déterminent au hasard les positions, les formes et les couleurs de deux nouvelles mouches et les dessinent.

290 appelle le sous-programme de fin de partie.

300 examine si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9290)

9020 détermine les couleurs.

9030 détermine la durée d'une partie et initialise le meilleur score.

9040-9150 fabriquent les caractères graphiques employés pour représenter 8 sortes de mouches.

9170-9210 fabriquent les caractères graphiques utilisés pour représenter l'araignée et son fil.

9230-9280 définissent les 3 fenêtres et en fixent les couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8100)

8020-8030 effacent l'écran et affichent le meilleur score.

8040-8060 dessinent le fil.

8070 fixe la position initiale de l'araignée.

8080 dessine l'araignée.

8090 initialise le score.

e) Le sous-programme ''mouche gobée'' (2000-2100)

2020 émet un son approprié.

2040-2070 déterminent le nombre de points correspondant à la mouche gobée.

2080-2090 actualisent le score et l'affichent.

f) Le sous-programme de fin de partie (7000-7110)

7020-7030 font retentir un petit accord.

7040 actualise le meilleur score.

7050-7060 attendent un instant avant d'effacer l'écran.

7070-7100 vous demandent si vous voulez rejouer.

5. Liste des variables

CF	Couleur de fond.
BR	Couleur du tour de l'écran.
CA	Couleur de l'araignée et de son fil.
CS	Couleur d'affichage des scores.
C3	Troisième couleur servant à représenter les mouches (les deux autres étant celle de l'araignée et celle des scores).
TM	Durée d'une partie.
MS	Meilleur score.
A$	Chaîne de deux caractères graphiques représentant l'araignée.
F$	Chaîne de deux caractères graphiques utilisés pour représenter le fil.
M$(8)	Tableau des caractères graphiques représentant les 8 mouches.
X	Position de l'araignée sur son fil.
SC	Score.
T	Compteur de temps (tours de boucle).
XN	Nouvelle position de l'araignée sur son fil.
X1,X2	Positions des deux dernières mouches (tirées au hasard).
PN1,PN2	Couleurs des deux dernières mouches (tirées au hasard).
N1,N2	Numéro de la « forme » des deux dernières mouches.

Variables auxiliaires

R$ A

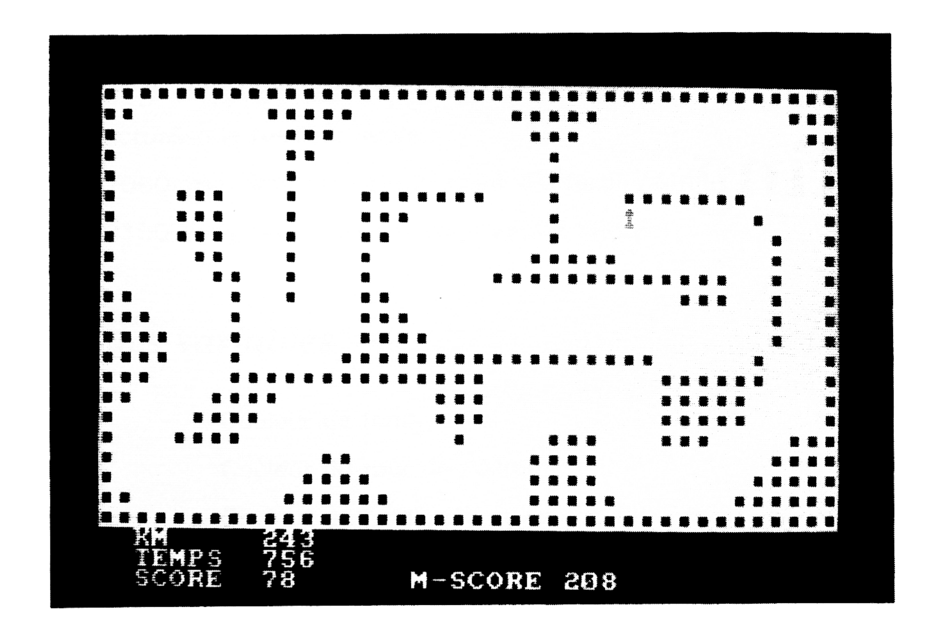
12
Karting

1. Le jeu

La piste est là, devant vos yeux, avec ses bottes de paille qui en délimitent le tracé. Votre bolide ne tarde pas à se présenter. Il vous suffit d'y monter et d'essayer de le conduire sur le plus grand nombre possible de kilomètres et aussi vite que vous le pouvez.

Le bolide apparaît en haut à gauche de l'écran. Le programme vous demande de signaler que vous êtes prêt en tapant sur une touche quelconque. Votre voiture se met alors en marche, à petite vitesse. A partir de cet instant, vous ne pourrez plus l'arrêter et la partie se terminera inexorablement... dans le décor.

Vous dirigez votre bolide à l'aide des quatre touches fléchées (ou, le cas échéant, avec une poignée de jeu). Vous accélérez à l'aide de la barre d'espace et vous freinez avec la touche ''TAB''. Le bruit du moteur vous renseignera sur votre vitesse.



A la fin de la partie (c'est-à-dire quand vous avez quitté la piste), le programme vous indique la distance parcourue et votre temps de parcours. Votre score est alors calculé en fonction de ces deux éléments (il est proportionnel à la distance et à la vitesse moyenne). Vous pouvez le comparer avec le meilleur score déjà réalisé.

2. *Le programme*

```
100 '********************** KARTING **************************
110 '   '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+9,16*(25-Y)+9)
130 GOSUB 9000
140  GOSUB 8000
150   R$=INKEY$: IF R$<>"" THEN GOSUB 2000
160   XN=X+(DN=3)-(DN=4): YN=Y+(DN=1)-(DN=2)
170   SOUND 1,200,1
180   IF FNSC(XN,YN)<>0 THEN GOSUB 3000: GOTO 260
190   LOCATE X,Y: PRINT V$(DN);
200   X=XN: Y=YN: DL=DN
210   KM=KM+1: T=T+DT
```

```
220   FOR K=1 TO AT: NEXT K
230   SOUND 1,200,1
240   GOTO 150
250  LOCATE X,Y: PRINT " ";
260 IF R$<>"N" THEN 140
270 END
280 '
2000 '------------ SP CHANGEMENT DE DIRECTION ----------------
2010 '
2020 DN=ASC(R$)-239
2030 IF DN=-230 OR DN=-207 THEN GOSUB 5000
2040 IF DN<1 OR DN>4 THEN DN=DL
2050 RETURN
2060 '
3000 '------------ SP SORTIE DE PISTE -----------------------
3010 '
3020 FOR H=10 TO 300 STEP 10: SOUND 1,H,1: NEXT H
3030 SOUND 1,0,20
3040 FOR I=1 TO 6
3050  SOUND 1,253,18: SOUND 1,319,18
3060  SOUND 1,253,18: SOUND 1,0,25
3070 NEXT I
3080 '
3090 LOCATE #1,3,1: PRINT #1,"KM     "; KM
3100 LOCATE #1,3,2: PRINT #1,"TEMPS "; T;
3110 SC=INT(KM*KM/T)
3120 LOCATE #1,3,3: PRINT #1,"SCORE "; SC;
3130 LOCATE #1,18,3: PRINT #1,"M-SCORE"; MS;
3140 IF SC>MS THEN MS=SC
3150 '
3160 FOR K=1 TO 200: NEXT K
3170 LOCATE X,Y: PRINT " ";
3180 WHILE INKEY$<>"": WEND
3190 LOCATE #1,20,1: PRINT #1,"rejouez vous(O/N)";
3200 R$=INKEY$: IF R$="" THEN 3200
3210 RETURN
3220 '
5000 '------------ SP CHANGEMENT DE VITESSE -----------------
5010 '
5020 DT=DT-(DN=-207)*(DT>0)+(DN=-230)*(DT<5)
5030 AT=KV*(DT-1)
5040 RETURN
5050 '
```

```
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 KM=0: T=0: DT=5: AT=KV*DT
8030 X=3: Y=4: DN=4: DL=DN
8040 LOCATE X,Y: PRINT CHR$(162);
8050 CLS #1
8060 LOCATE #1,2,2
8070 PRINT #1, "pour commencer, tapez sur une touche";
8080 R$=INKEY$: IF R$="" THEN 8080
8090 CLS #1
8100 SOUND 1,200,1
8110 RETURN
8120 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 CB=1: CP=6: CF=24: CS=24: CV=6
9030 MS=0
9040 KV=50
9050 '
9060 SYMBOL AFTER 160
9070 SYMBOL 160,0,0,60,60,60,60,0,0
9080 SYMBOL 161,0,0,99,255,255,99,0,0
9090 SYMBOL 162,0,0,198,255,255,198,0,0
9100 SYMBOL 163,60,60,24,24,24,60,60,24
9110 SYMBOL 164,24,60,60,24,24,24,60,60
9120 DIM V$(4)
9130 V$(1)=" "+CHR$(11)+CHR$(8)+CHR$(164)
9140 V$(2)=" "+CHR$(10)+CHR$(8)+CHR$(163)
9150 V$(3)=CHR$(8)+CHR$(161)+" "
9160 V$(4)=" "+CHR$(162)
9170 B$=CHR$(160)
9180 '
9190 INK 0,CF: INK 1,CB: INK 2,CV: INK 3,CS
9200 CLS: PAPER 0: BORDER CB
9210 WINDOW #1,1,40,23,25: PAPER #1,1: PEN #1,0
9220 '
9230 PEN 1
9240 FOR I=1 TO 22
9250  READ CH$
9260  FOR J=1 TO 40
9270   C$=MID$(CH$,J,1)
9280   IF C$<>" " THEN LOCATE J,I: PRINT B$;
9290  NEXT J
9300 NEXT I
9310 '
9320 PEN 2
9330 RETURN
```

```
9340 '
9350 DATA XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
9360 DATA XX       XXXXX        XXXXX          XXX
9370 DATA X         XXX          XXX            XX
9380 DATA X         XX            X              X
9390 DATA X         X             X              X
9400 DATA X   XXX   X   XXXXXXX   X   XXXXXXX    X
9410 DATA X   XXX   X   XXX       X          X   X
9420 DATA X   XXX   X   XX        X           X  X
9430 DATA X    XX   X   X        XXXXX        X  X
9440 DATA X     XX  X   X      XXXXXXXXXXXXX  X  X
9450 DATA XX     X  X   XX               XXX  X  X
9460 DATA XXX    X      XXX                   X  X
9470 DATA XXXX   X      XXXX                  X  X
9480 DATA XXXX   X     XXXXXXXXXXXXXXXXX     X   X
9490 DATA XXX    XXXXXXXXXXXXXX         XXXXXX   X
9500 DATA XX    XXXX        XXX         XXXXX    X
9510 DATA X    XXXX         XXX         XXXXX    X
9520 DATA X   XXXX           X    XXX   XXX    XXX
9530 DATA X           XX         XXXX         XXXX
9540 DATA X          XXXX        XXXX        XXXXX
9550 DATA XX        XXXXXX       XXXXX      XXXXXX
9560 DATA XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur de la piste (ici jaune)**

Remplacez, en 9020, la valeur 24 (dans CF = 24) par un nombre de votre choix, entre 0 et 26.

▷ **Pour modifier la couleur des "bottes de paille"**

Remplacez, en 9020, la valeur 6 (dans CP = 6) par un nombre de votre choix, entre 0 et 26.

▷ **Pour modifier le tracé de la piste**

Il vous suffit de modifier les instructions "DATA" des lignes 9350 à 9560 en tenant compte des remarques suivantes :

- Chacune de ces 22 instructions représente une ligne de la piste.

- Chaque instruction contient une chaîne d'au maximum 40 caractères et formée uniquement de caractères ''espace'' et ''X''. Les espaces matérialisent la piste tandis que ''X'' correspond aux ''bottes de paille''.
- Il est absolument indispensable que la piste soit toujours bordée de ''X''. En effet, pour obtenir la vitesse maximum, le programme ne teste pas les ''débordements'' d'écran ; il se contente de vérifier s'il y a ou non collision avec une botte de paille. Dans ces conditions, l'absence de ''X'' en bord de l'écran pourrait entraîner un mauvais fonctionnement du programme.

▷ **Si le jeu vous paraît trop rapide ou trop lent**

Vous diminuerez les différentes vitesses (à l'exception de la cinquième), en remplaçant, en 9040, la valeur 50 par un nombre plus grand. Notez que si vous placez là une valeur inférieure à 50, le jeu deviendra plus rapide.

4. *Description du programme*

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Lecture du clavier,
- Examen du contenu de l'écran.

b) Le programme principal (100-270)

120 définit une fonction permettant de connaître le numéro d'encre d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-260 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150-240 sont répétées jusqu'à ce que la voiture quitte la piste :

150 appelle le sous-programme de ''changement'' si une touche du clavier a été pressée. Celui-ci gère à la fois le changement de vitesse et le changement de direction.

160 détermine la nouvelle position de la voiture.

170 émet un son.

180 examine si la voiture sort de la piste ; dans ce cas le sous-programme de sortie de piste est appelé et la partie est interrompue.

190-200 déplacent la voiture. Notez que la chaîne V$(DN) contient à la fois le caractère d'effacement de la voiture, les déplacements de curseur et la voiture dans sa nouvelle position.

210 actualise le temps et le kilométrage.

220 réalise une attente fonction de la vitesse.

230 émet un son.

250 efface la voiture (cas de fin de partie).

260 examine si vous voulez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9550)

9020 détermine les couleurs.

9030 initialise le meilleur score.

9040 fixe le coefficient employé dans les ''boucles d'attente''.

9070-9170 déterminent les caractères graphiques représentant les bottes de paille et la voiture dans chacune des quatre directions de déplacement.

9190-9210 définissent la fenêtre numéro 1 et fixent les différentes couleurs.

9230-9300 dessinent la piste en utilisant les instructions DATA des lignes 9350-9560. Notez que chaque caractère ''X'' correspond à une ''botte de paille''.

d) Le sous-programme d'initialisation d'une partie (8000-8110)

8020 initialise les compteurs ''temps'' et ''kilomètres'' et fixe la valeur initiale du temps d'attente (conditionnant la vitesse de la voiture).

8030 fixe la position initiale de la voiture et sa direction de déplacement.

8040 dessine la voiture dans sa position initiale.

8050 efface la partie inférieure de l'écran.

8060-8100 attendent que vous tapiez sur une touche.

e) Le sous-programme de changement de direction (2000-2050)

2020-2030 appellent, s'il y a lieu, le sous-programme de changement de vitesse.

2040 vérifie que la touche pressée est bien l'une des quatre touches fléchées et, dans ce cas, actualise la direction de déplacement de la voiture.

f) Le sous-programme de changement de vitesse (5000-5030)

Il calcule le nouveau temps d'attente.

g) Le sous-programme de sortie de piste (3000-3210)

3020-3030 émettent un son approprié.

3040-3070 font retentir un avertisseur d'ambulance.

3090-3100 affichent le nombre de kilomètres parcourus et le temps.

3110-3140 calculent le score, l'affichent en même temps que le nouveau score et actualisent ce dernier.

3160-3200 après une légère attente, effacent la voiture et vous demandent si vous souhaitez rejouer.

5. Liste des variables

CB	Couleur de la bordure de l'écran et du fond de la fenêtre 1.
CP	Couleur des ''bottes de paille''.
CF	Couleur de la piste et des scores.
CV	Couleur de la voiture.
MS	Meilleur score.
KV	Coefficient employé pour définir un temps d'attente (AT) en fonction de la vitesse.
BL$	Chaîne de 38 caractères ''espace'' employée pour effacer des messages.
B$	Caractère graphique représentant une ''botte de paille''.
V$(4)	Tableau des chaînes utilisées pour ''effectuer'' le déplacement de la voiture. Chaque chaîne contient à la fois un espace servant à effacer la voiture de son ancienne position, les déplacements de curseur et la voiture dans sa nouvelle position. ● V$(1) correspond à la voiture dirigée vers le haut. ● V$(2) correspond à la voiture dirigée vers le bas,

- V$(3) correspond à la voiture dirigée vers la gauche,
- V$(4) correspond à la voiture dirigée vers la droite.

KM Kilomètres parcourus depuis le début de la partie.

T Temps écoulé depuis le début de la partie.

DT Représente la vitesse de la voiture. (1 correspond à la vitesse la plus élevée, 5 à la vitesse la plus basse).

AT Temps d'attente dépendant de la vitesse (il est proportionnel à la valeur de DT).

X,Y Coordonnées de la voiture.

DN Code du déplacement à venir de la voiture.
(1 : ↑ ; 2 : ↓ ; 3 : ← ; 4 : →).

DL Code de l'ancien déplacement de la voiture.

XN,YN Coordonnées du prochain emplacement de la voiture.

SC Score.

Variables auxiliaires

I J

CH$ C$

R$

13
Tir aux ballons

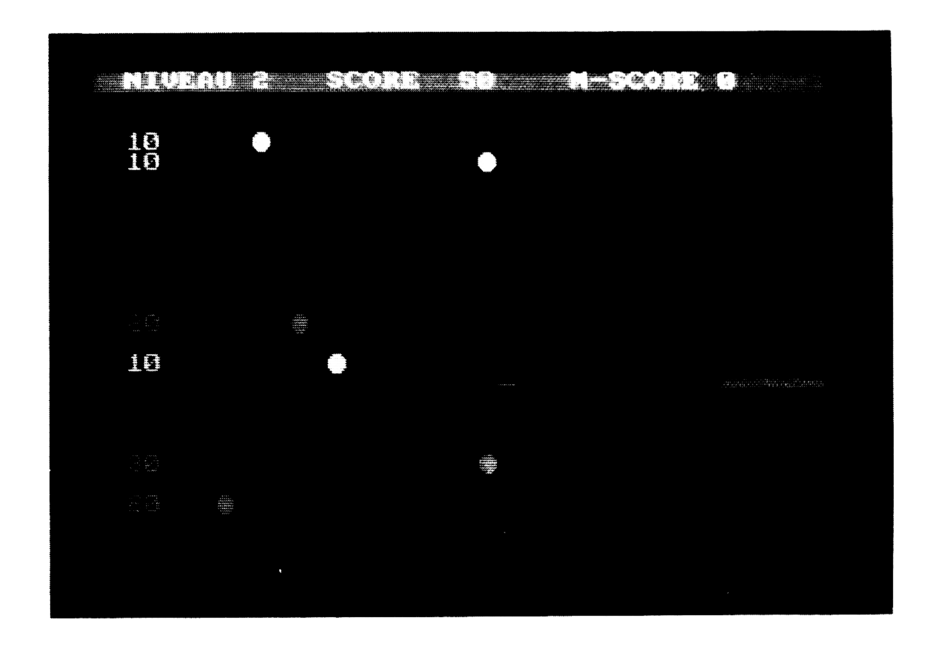
1. Le jeu

Venez pour un moment vous distraire à la fête foraine en montrant vos talents de tireur. Une myriade de ballons multicolores s'élève devant vos yeux. Crevez-en le plus possible avec votre sarbacane.

La sarbacane apparaît sur le côté droit de l'écran et se déplace verticalement à l'aide des deux touches " ↑ " et " ↓ ". Vous tirez une flèche en appuyant sur la barre d'espace. Tant qu'elle n'a pas traversé l'écran ou crevé un ballon, vous ne pouvez pas en tirer une nouvelle.

Trois niveaux de jeu vous sont proposés :

- *Niveau 1 :* Tous les ballons rapportent le même nombre de points (10).
- *Niveau 2 :* Le nombre de points dépend de la couleur du ballon : 10 pour un jaune, 20 pour un bleu, 30 pour un rose. Ce nombre apparaît sur la gauche du ballon.



- *Niveau 3 :* Les ballons deviennent capricieux : lorsqu'ils sont touchés pour la première fois, ils se contentent de diminuer de taille. Il vous faut les toucher une seconde fois pour les faire disparaître et marquer les points correspondants.

La partie dure un temps déterminé à l'avance. Le programme affiche en permanence, en haut de l'écran, votre score et le meilleur score du niveau auquel vous jouez.

2. Le programme

```
100 '*************** TIR AUX BALLONS **************************
110 '
120 DEF FNSC(X,Y) = TEST (16*(X-1)+7,16*(25-Y)-8)
130 GOSUB 9000
140  GOSUB 8000
150  FOR T=1 TO TM
160   GOSUB 3000: GOSUB 4000: GOSUB 3000: GOSUB 5000
170   GOSUB 3000: GOSUB 4000: GOSUB 3000
180  NEXT T
190  IF SC>MS(NV) THEN MS(NV)=SC
```

```
200  FOR K=1 TO 400: NEXT K
210  CLS #1
220  WHILE INKEY$<>"": WEND
230  LOCATE #1, 7,25
240  PRINT #1, "voulez vous rejouer (O/N)?"
250  R$=INKEY$: IF R$="" THEN 250
260 IF R$<>"N" THEN 140
270 END
280 '
3000 '------------ SP DEPLACEMENT FLECHE --------------------
3010 '
3020 IF XF=0 THEN RETURN
3030 IF FNSC(XF-1,YF)<>0 THEN 3070
3040  LOCATE #1, XF,YF: PRINT #1, " ";: XF=XF-2.
3050  IF XF<X1 THEN XF=0 ELSE LOCATE #1,XF,YF: PRINT #1,F$;
3060  RETURN
3070 SOUND 1,478,2
3080 LOCATE #1, XF,YF: PRINT #1, " ";
3090 LOCATE #1, XF-1,YF: C=TEST(16*(XF-2)+3,16*(25-YF)-9)
3100 IF C<>0 AND NV=3 THEN PEN #1,C: PRINT#1,B2$;:XF=0:RETURN
3110 SC=SC+10*TEST(16*(XF-2)+7,16*(25-YF)-9)
3120 PRINT #1," ";
3130 LOCATE 20,1: PRINT SC;
3140 XF=0
3150 RETURN
3160 '
4000 '------------ SP DEPLACEMENT SARBACANE ET TIR ----------
4010 '
4020 IF INKEY(47)<>0 OR XF<>0 THEN 4060
4030  XF=XS-2: YF=YS
4040  PEN #1,2: LOCATE #1,XF,YF: PRINT #1, F$;
4050  RETURN
4060 PEN #1,2: LOCATE #1, XS,YS: PRINT #1,B$;
4070 YS=YS+(INKEY(8)=0)*(YS<23)-(INKEY(1)=0)*(YS>1)
4080 LOCATE #1, XS,YS: PRINT #1, S$;
4090 RETURN
4100 '
5000 '------------ SP DEPLACEMENT BALLONS --------------------
5010 '
5020 IF RND(1)>ZB THEN 5070
5030  X=H(K): CB=H(K+1): K=K+2: IF K>NX THEN K=1
5040  IF NV=1 THEN PT=10 ELSE PT=10*CB
5050  PEN #1,CB: LOCATE #1,2,24: PRINT #1, PT;
5060  LOCATE #1, X,24: PRINT #1, B1$;
```

```
5070 LOCATE #1, XS,YS: PRINT #1, B$;
5080 IF XF<>0 THEN LOCATE #1, XF,YF: PRINT #1, " ";
5090 LOCATE #1, 1,24: PRINT #1,CHR$(10)
5100 PEN #1,2: LOCATE #1, XS,YS: PRINT #1, S$;
5110 IF XF=0 THEN RETURN
5120 IF FNSC(XF,YF)<>0 THEN XF=0 ELSE LOCATE#1,XF,YF:PRINT#1,F$;
5130 RETURN
5140 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 PEN #1,2: CLS #1
8030 LOCATE #1,9,15
8040 PRINT #1, "niveau choisi (1 a 3)";
8050 R$=INKEY$: IF R$="" THEN 8050
8060 NV=VAL(R$): IF NV<1 OR NV>3 THEN 8050
8070 PRINT #1, NV;
8080 '
8090 LOCATE #1, 12,24: PRINT #1, "PATIENTEZ, SVP";
8100 FOR K=1 TO NX-1 STEP 2
8110  H(K)=X1+2*INT(RND(1)*(X2-X1)/2)
8120  CB=INT(RND(1)*3+1)
8130  H(K+1)=CB
8140 NEXT K
8150 '
8160 SC=0: XF=0: YS=15: K=1
8170 CLS #1
8180 LOCATE 3,1: PRINT "NIVEAU"; NV;
8190 LOCATE 14,1: PRINT "SCORE        ";
8200 LOCATE 27,1: PRINT "M-SCORE"; MS(NV);
8210 RETURN
8220 '
9000 '------------ SP INITIALISATION DU JEU-------------------
9010 '
9020 CF=0: FS=16: CA=24: CS=7: BR=0
9030 X1=8: X2=26
9040 XS=35: LS=6: XF=0: ZB=0.2
9050 TM=10: NX=100
9060 DIM MS(3), H(NX)
9070 '
9080 SYMBOL AFTER 160
9090 SYMBOL 160,0,0,255,255,255,255,0,0
9100 SYMBOL 161,0,0,0,0,255,0,0,0
9110 SYMBOL 162,60,126,255,255,255,255,126,60
9120 SYMBOL 163,0,0,24,60,60,24,0,0
```

```
9130 S$="": FOR I=1 TO LS: S$=S$+CHR$(160): NEXT I
9140 B$="": FOR I=1 TO LS: B$=B$+" ": NEXT I
9150 F$=CHR$(161): B1$=CHR$(162): B2$=CHR$(163)
9160 '
9170 INK 0,CF: INK 1,CA: INK 2,CS: INK 3,FS
9180 WINDOW #1,1,40,2,25
9190 PAPER 3: PEN 1: BORDER BR: CLS
9200 PAPER #1,0: CLS #1
9210 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la durée d'une partie**

Remplacez, en 9050, la valeur 300 (dans TM = 300), par un nombre de votre choix. Par exemple, 600 conduira à des parties deux fois plus longues tandis que 100 conduira à des parties trois fois plus courtes.

▷ **Pour que les ballons soient plus ou moins nombreux**

Remplacez, en 9040, la valeur 0.2 (dans ZB = 0.2) par un nombre compris entre 0 et 1 sachant que :

0.2 correspond, en moyenne, à un ballon toutes les cinq lignes.

0.5 à un ballon toutes les deux lignes.

1 à un ballon sur chaque ligne.

▷ **Pour utiliser une poignée de jeu :**

Remplacez la ligne 4020 par :

```
4020 D = JOY(0) : IF (D AND 16)= 0 OR XF <> 0 THEN 4060
```

et la ligne 4070 par :

```
4070 YS = YS + (D=1) * (YS <23) - (D=2) * (YS >1)
```

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard,

— Animation,

— Lecture rapide du clavier,

— Examen du contenu de l'écran.

b) Le programme principal (100-270)

120 définit une fonction permettant de connaître la couleur (numéro d'encrier) d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-260 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150 et 180 répètent 300 fois les instructions 160 et 170 qui appellent les sous-programmes :

- déplacement de la flèche (3000)
- déplacement de l'ensemble des ballons (5000)
- déplacement de la sarbacane (4000)

190 actualise le score.

200-250 après une brève attente, vous demandent si vous souhaitez rejouer.

240 examine votre réponse.

c) Le sous-programme d'initialisation du jeu (9000)

9020 détermine les couleurs.

9030 détermine le domaine dans lequel évoluent les ballons.

9040-9050 fixent la position initiale de la sarbacane, sa longueur, la fréquence d'apparition des ballons, la durée de la partie.

9080-9150 fabriquent les caractères graphiques représentant la sarbacane, la flèche et les ballons.

9170-9200 définissent la fenêtre 1 et fixent les couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8210)

8020-8070 vous font choisir votre niveau de jeu, en rejetant les réponses autres que 1, 2 ou 3.

8090-8140 vous demandent de patienter et préparent un tableau de valeurs tirées au hasard qui serviront à déterminer la position et la couleur de chaque ballon.

(Cette façon de procéder permet de ''gagner'' un petit peu de temps pendant le déroulement du jeu).

8160 initialise le score, la position de la sarbacane.

8170-8200 effacent l'écran et affichent le niveau, le score et le meilleur score.

e) Le sous-programme de déplacement de la flèche (3000-3150)

3020 examine si une flèche a été tirée. Si c'est le cas :

3030 examine si la flèche peut être déplacée sans toucher un ballon :

3040-3050 déplacent la flèche lorsque cela est possible en vérifiant qu'elle n'atteint pas le bord gauche.

3070-3140 correspondent au cas où un ballon est touché :

3070-3080 émettent un son approprié et effacent la flèche.

3090-3100 examinent si le ballon est de taille normale ; dans ce cas, si l'on est en niveau 3, il est remplacé par un plus petit.

3110-3140 incrémentent le score et l'affichent, effacent le ballon touché et mettent à zéro l'indicateur ''flèche en cours''.

f) Le sous-programme de déplacement de la sarbacane et de tir (4000-4090)

4020 ''lit'' le clavier.

4030-4050 placent une flèche en position initiale si la barre d'espace a été pressée et qu'aucun tir n'est en cours.

4060-4080 déplacent la sarbacane.

g) Le sous-programme de déplacement de l'ensemble des ballons (5000-5130)

5020-5060 gèrent l'apparition des nouveaux ballons :

5020 décide de l'éventuelle apparition d'un nouveau ballon.

5030 en détermine l'abscisse et la couleur.

5040 détermine le nombre de points correspondants.

5050-5060 affichent le nombre de points et le ballon.

5070-5080 effacent la sarbacane et la flèche (si elle existe).

5090 provoque le défilement d'écran (scroll).

5100-5120 redessinent la sarbacane et, s'il y a lieu, la flèche.

5. Liste des variables

CF	Couleur du fond de l'écran.
FS	Couleur de fond de la ligne supérieure où s'affichent les scores.
CA	Couleur d'affichage des scores.
CS	Couleur de la sarbacane et de la flèche.
BR	Couleur de la bordure de l'écran.
X1,X2	Bornes de l'intervalle à l'intérieur duquel pourront se trouver les abscisses des ballons (ces valeurs doivent impérativement être paires).
XS,YS	Coordonnées de l'extrémité gauche de la sarbacane.
LS	Longueur de la sarbacane.
XF,YF	Coordonnées de la flèche. (Si XF = 0, aucune flèche n'est présente sur l'écran).
ZB	Probabilité d'apparition d'un ballon sur chaque ligne.
TM	Durée d'une partie.
NX	Nombre de valeurs qui seront tirées au hasard avant le début d'une partie.
MS(3)	Tableau des meilleurs scores relatifs à chaque niveau.
H(NX)	Tableau destiné à contenir les valeurs tirées au hasard en début de partie.
S$	Chaîne de caractères graphiques représentant la sarbacane.
B$	Chaîne de caractères "espace" utilisée pour l'effacement de la sarbacane.
F$	Caractère graphique représentant la flèche.
B1$	Caractère graphique représentant un ballon de taille normale.
B2$	Caractère graphique représentant un ballon "rétréci".
NV	Niveau de jeu.
SC	Score.
K	Pointeur dans le tableau H.
T	Temps écoulé (compteur de boucle).

X	Abscisse d'un nouveau ballon.
CB	Couleur d'un nouveau ballon.

Variables auxiliaires

R$ A C

K

14
Mur de briques

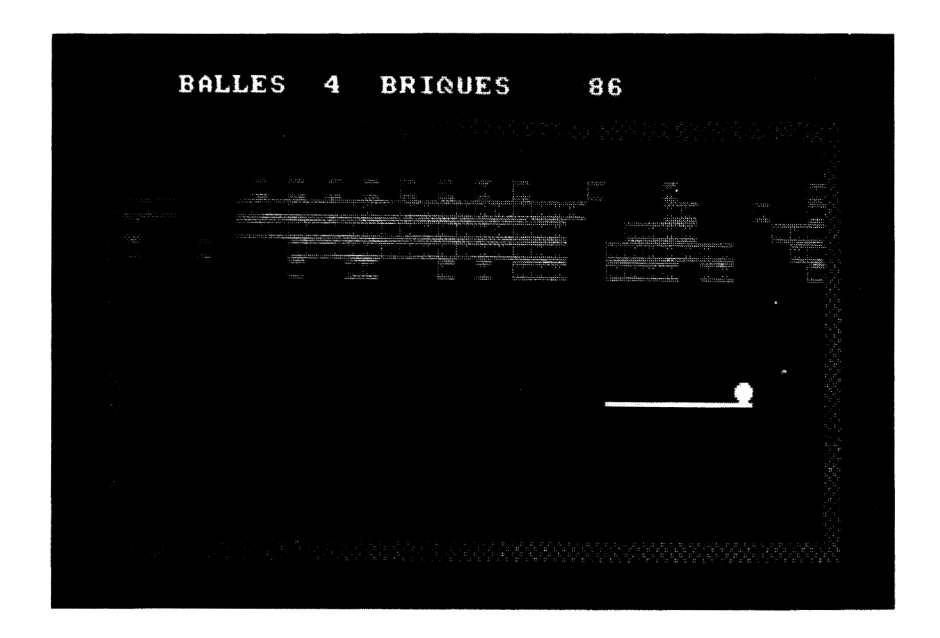
1. Le jeu

C'est là un jeu très classique que vous avez certainement déjà rencontré. A l'intérieur d'un cadre se trouve un mur de briques. Une balle se promène sur l'écran, rebondissant sur le cadre ou sur une ''raquette mobile''. Elle rebondit également sur les briques mais en les détruisant du même coup.

Par rapport aux versions habituelles de ce jeu, nous avons introduit une nouveauté : la raquette peut se déplacer à la fois horizontalement et verticalement (dans certaines limites).

L'objectif est de détruire le maximum de briques dans un minimum de temps. Pouvoir rapprocher la raquette du mur vous fera donc gagner du temps et par suite améliorer votre score.

Chaque balle ratée par la raquette est perdue. Vous disposez de six balles en tout.



La raquette se déplace à l'aide des quatre touches fléchées ou, le cas échéant, de la poignée de jeu.

Le programme affiche en permanence le nombre de balles restantes et le nombre de briques détruites. La partie s'achève, soit quand vous n'avez plus de balles, soit (ce qui est plus rare) quand le mur a été entièrement détruit. Le programme vous fait alors connaître votre score qui est calculé en fonction du nombre de briques détruites et du temps mis pour y parvenir. En même temps, vous pourrez comparer votre résultat avec le meilleur score déjà réalisé.

2. Le programme

```
100 '****************** MUR DE BRIQUES **********************
110 '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+1,16*(25-Y)+15)
130 GOSUB 9000
140  GOSUB 8000: GOSUB 7000
150  GOSUB 4000: GOSUB 5000: GOSUB 4000: T=T+1
160  IF ND<NB AND NC>0 THEN 150
170  GOSUB 8500
180 IF R$<>"N" THEN 140
190 END
200 '
```

```
4000 '------------ SP DEPLACEMENT BALLE ---------------------
4010 '
4020 C1=FNSC(X,Y+DY): C2=FNSC(X+DX,Y)
4030 C3=FNSC(X+DX,Y+DY)
4040 IF C1<>0 OR C2<>0 OR C3<>0 THEN GOSUB 6000: RETURN
4050  LOCATE X,Y: PRINT " ";
4060  X=X+DX: Y=Y+DY
4070  IF Y<YR THEN LOCATE X,Y: PRINT BA$;: RETURN
4080 GOSUB 6500
4090 RETURN
4100 '
5000 '------------ SP DEPLACEMENT RAQUETTE -------------------
5010 '
5020 RG=INKEY(8): RD=INKEY(1): RH=INKEY(0): RB=INKEY(2)
5030 IF RG*RD*RH*RB<>0 THEN RETURN
5040 LOCATE XR,YR: PRINT B$;
5050 XR=XR+2*((RD=0)*(XR<XM-1)-(RG=0)*(XR>X1+1))
5060 YR=YR+(RB=0)*(YR<R2)+(RH=0)*(YR>R1)*(YR<>Y+1)
5070 LOCATE XR,YR: PRINT RA$;
5080 RETURN
5090 '
6000 '------------ SP REBOND ---------------------------------
6010 '
6020 SOUND 1,478,6
6030 IF C1=0 THEN 6090
6040  IF C1<>3 THEN 6060
6050   LOCATE X,Y+DY: PRINT " ";: GOSUB 6800
6060  DY=-DY
6070  RETURN
6080 '
6090 IF C2=0 THEN 6150
6100  IF C2<>3 THEN 6120
6110   LOCATE X+DX,Y: PRINT " ";: GOSUB 6800
6120  DX=-DX
6130  RETURN
6140 '
6150  IF C3<>3 THEN 6170
6160   LOCATE X+DX,Y+DY: PRINT " ";: GOSUB 6800
6170  DX=-DX: DY=-DY
6180  IF RND(1)>0.75 THEN DX=-DX
6190 RETURN
6200 '
```

```
6500 '------------ SP PERTE BALLE ----------------------------
6510 '
6520 NC=NC-1
6530 LOCATE #1,12,1: PRINT #1, NC;
6540 FOR H=20 TO 300 STEP 5: SOUND 1,H,1: NEXT H
6550 FOR K=1 TO 500: NEXT K
6560 IF NC>0 THEN GOSUB 7000
6570 RETURN
6580 '
6800 '------------ SP BRIQUE DETRUITE ------------------------
6810 '
6820 ND=ND+1
6830 LOCATE #1, 26,1: PRINT #1, ND;
6840 RETURN
6850 '
7000 '------------ SP PLACEMENT HASARD BALLE -----------------
7010 '
7020 X=X1+1+INT(RND(1)*(X2-X1-1))
7030 Y=Y4+2
7040 LOCATE X,Y: PRINT BA$;
7050 IF RND(1)>0.5 THEN DX=1 ELSE DX=-1
7060 IF RND(1)>0.5 THEN DY=1 ELSE DY=-1
7070 IF NC=NX THEN 7090
7080  LOCATE XR,YR: PRINT B$;
7090 XR=INT((X1+X2-LR)/2): YR=R2
7100 LOCATE XR,YR: PRINT RA$;
7110 RETURN
7120 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 T=0: ND=0: NC=NX
8030 CLS: CLS #1
8040 LOCATE #1, 5,1: PRINT #1, "BALLES     BRIQUES";
8050 LOCATE #1, 12,1: PRINT #1, NC;
8060
8070 PEN 2
8080 FOR X=X1 TO X2
8090  LOCATE X,Y1: PRINT M$;
8100  LOCATE X,Y2: PRINT M$;
8110 NEXT X
8120 FOR Y=Y1 TO Y2
8130  LOCATE X1,Y: PRINT M$;
8140  LOCATE X2,Y: PRINT M$;
8150 NEXT Y
```

```
8160 '
8170 PEN 3
8180 FOR X=X1+1 TO X2-1
8190  FOR Y=Y3 TO Y4
8200   LOCATE X,Y: PRINT BR$;
8210  NEXT Y
8220 NEXT X
8230 '
8240 PEN 1
8250 RETURN
8260 '
8500 '------------ SP FIN DE PARTIE --------------------------
8510 '
8520 SC=INT(100*ND*ND/T*NX/(NX-NC))
8530 LOCATE 6,13: PRINT "SCORE"; SC;
8540 LOCATE 6,15: PRINT "M-SCORE"; MS;
8550 IF SC>MS THEN MS=SC
8560 WHILE INKEY$<>"": WEND
8570 LOCATE 6,18: PRINT "voulez vous rejouer (O/N)";
8580 R$=INKEY$: IF R$="" THEN 8580
8590 RETURN
8600 '
9000 '------------ SP INITIALISATION DU JEU ------------------
9010 '
9020 MS=0: NX=6
9030 CE=0: BR=0: CA=6: CB=7: CR=24
9040 X1=1: X2=40: Y1=3: Y2=24
9050 Y3=6: Y4=10: R1=13: R2=23: LR=8
9060 XM=X2-LR
9070 NB=(X2-X1-1)*(Y4-Y3+1)
9080 '
9090 SYMBOL AFTER 160
9100 SYMBOL 160, 204,204,51,51,204,204,51,51
9110 SYMBOL 161, 255,255,168,128,128,168,255,255
9120 SYMBOL 162, 60,126,255,255,255,255,126,60
9130 SYMBOL 163,255,255,0,0,0,0,0,0
9140 M$=CHR$(160): BR$=CHR$(161): BA$=CHR$(162)
9150 RA$="": FOR I=1 TO LR: RA$=RA$+CHR$(163): NEXT I
9160 B$="": FOR I=1 TO LR: B$=B$+" ": NEXT I
9170 '
9180 INK 0,CE: INK 1,CR: INK 2,CA: INK 3,CB
9190 WINDOW #1,1,40,1,Y2-1
9200 PAPER 0: CLS : BORDER BR
9210 PAPER #1,0: PEN #1,1: CLS #1
9220 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

◁ **Pour modifier le nombre de balles d'une partie**

Remplacez, en 9020, le nombre 6 (dans NX = 6) par une valeur de votre choix.

◁ **Pour modifier la taille ou la position du mur de briques**

Il vous suffit de savoir que, en 9050, les valeurs de Y3 (ici 6) et de Y4 (ici 10) correspondent aux numéros des lignes d'écran entre lesquelles s'étend le mur.

Ainsi, par exemple :

```
9050 Y3=6 : Y4=11 : R1=13 : R2=23 : LR=8
```

vous fournira un mur commençant toujours à la 6e ligne, mais s'étendant jusqu'à la 11e ligne.

De même :

```
9060 Y3=7 : Y4=11 : R1=13 : R2=23 : LR=8
```

vous fournira un mur de même taille que le mur d'origine, mais situé un peu plus bas sur l'écran.

Attention : la valeur de Y3 doit toujours rester strictement supérieure à celle de Y1, définie en ligne 9040 (actuellement 3). La valeur de Y4 doit toujours être strictement inférieure à celle de R1, définie en ligne 9050 (actuellement 13).

◁ **Pour modifier les limites du domaine où se déplace la raquette**

Il vous suffit de savoir que, en 9050, les valeurs de R1(13) et de R2(23) en fixent les limites verticales.

Par exemple :

```
9050 Y3=5 : Y4=6 : R1=15 : R2=22 : LR=8
```

vous conduira à un domaine plus restreint.

Avec :

```
9050 Y3=6 : Y4=10 : R1=22 : R2=22 : LR=8
```

vous obtiendrez un jeu où la raquette ne peut plus se déplacer verticalement (vous retrouverez la version classique du ''mur de briques'').

◁ **Pour modifier la taille de la raquette**

Remplacez, en 9050, le nombre 8 (dans LR = 8) par une valeur de votre choix.

◁ **Pour utiliser une poignée de jeu**

Remplacez les lignes 5020 et 5030 par :

```
5020 RJ=JOY(0) : RG=RJ AND 4 : RD=RJ AND 8
5030 RB=RJ AND 2 : RH=RJ AND 1 : IF RG+RD+RH+RB=0 THEN RETURN
```

et les lignes 5050 et 5060 par :

```
5050 XR=XR+2 * ((RD=8) * (XR <XM-1) - (RG=4) * (XR>X1 +1))
5060 YR=YR+(RB=2) * (YR<R2) + (RH=1) * (YR >R1) * (YR < > Y + 1)
```

Notez que les déplacements ''en diagonale'' sont ainsi autorisés.

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard,
— Animation,
— Lecture rapide du clavier,
— Examen du contenu de l'écran.

b) Le programme principal (100-190)

120 définit une fonction permettant de connaître la couleur d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-180 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle les sous-programmes d'initialisation d'une partie et de placement d'une balle au hasard.

150-160 sont répétées jusqu'à ce que toutes les briques soient détruites ou que toutes les balles aient été utilisées :

150 appelle les sous-programmes de déplacement de la balle et de la raquette et incrémente le compteur de temps.

160 examine si la partie est terminée.

170 appelle le sous-programme de fin de partie.

180 examine si vous voulez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9220)

9020 initialise le meilleur score et fixe le nombre de balles.

9030 détermine les couleurs.

9040-9070 fixent les coordonnées du cadre et du domaine où évolue la raquette, la longueur de la raquette et calculent le nombre de briques du mur.

9090-9160 fabriquent les chaînes de caractères graphiques représentant le mur, le cadre, la balle et la raquette ainsi que la chaîne destinée à l'effacement de la raquette.

9180-9210 définissent la fenêtre numéro 1 et fixent les couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8250)

8020 initialise les compteurs de temps, de briques détruites et de balles restantes.

8030-8050 affichent le nombre de balles.

8060-8150 dessinent le cadre.

8160-8220 dessinent le mur de briques.

e) Le sous-programme de placement d'une balle au hasard (7000-7110)

7020-7040 déterminent au hasard les coordonnées de la balle et la dessinent.

7050-7060 déterminent au hasard la direction de déplacement de la balle.

7070-7080 effacent la raquette de son ancienne position (sauf si l'on est en début de partie).

7090-7100 redessinent la raquette en "position initiale".

f) Le sous-programme de déplacement de la balle (4000-4090)

Pour chaque déplacement, il considère les couleurs C1, C2, C3 (obtenues par TEST) de trois cases, comme l'indiquent ces schémas, correspondants aux 4 directions possibles de la balle.



4020-4030 déterminent le contenu des trois cases C1, C2 et C3.

4040 cas où l'une de ces cases est occupée : on appelle le sous-programme de rebond.

4050-4080 cas où les trois cases sont disponibles :

4050 efface la balle.

4060 détermine les nouvelles coordonnées de la balle.

4070 dessine la balle après s'être assurée que la raquette ne la ratait pas.

4080 appelle le sous-programme ''perte de balle'' lorsque la raquette perd la balle.

g) Le sous-programme de perte de la balle (6500-6570)

6520-6530 actualisent le nombre de balles restantes.

6540 émet un son approprié.

6560 appelle, lorsqu'il reste encore des balles à jouer, le sous-programme de placement d'une balle au hasard.

h) Le sous-programme de rebond (6000-6190)

6020 émet un son approprié.

6030-6070 correspondent au cas où C1 est occupée :

6040-6050 examinent s'il s'agit d'une brique et appellent, dans ce cas, le sous-programme ''brique détruite'' qui actualise le compteur de briques détruites.

6060 assure, dans tous les cas (bord ou brique) le rebond de la balle.

6090-6130 correspondent au cas où C2 est occupée (sans que C1 le soit) :

6100-6110 examinent s'il s'agit d'une brique et appellent dans ce cas, le sous-programme ''brique détruite''.

6120 assure, dans tous les cas, le rebond de la balle.

6150-6180 correspondent au cas où seule C3 est occupée :

6160 examine s'il s'agit d'une brique et appelle, dans ce cas, le sous-programme ''brique détruite''.

6170 assure, dans tous les cas, le rebond de la balle.

6180 modifie, aléatoirement, le rebond (on est ici dans le cas où l'on touche une brique par un coin).

i) Le sous-programme de déplacement de la raquette (5000-5090)

5020 ''lit'' le clavier.

5030 examine s'il y a lieu de déplacer la raquette.

5040 efface la raquette.

5050-5070 déterminent la nouvelle position de la raquette et la dessinent.

j) Le sous-programme de fin de partie (8500-8590)

8520-8550 calculent le score et l'affichent, en même temps que le meilleur score.

8560 actualise le meilleur score.

8570-8580 vous demandent si vous voulez rejouer.

5. Liste des variables

MS	Meilleur score.
NX	Nombre total de balles.
CE	Couleur du fond de l'écran.
BR	Couleur du bord de l'écran.
CR	Couleur de la raquette, de la balle et des scores.
CA	Couleur du cadre.
CB	Couleur des briques.
X1,X2,Y1,Y2	Coordonnées du cadre.
Y3,Y4	Limites (verticales) entre lesquelles s'étend le mur de briques.
R1,R2	Limites (verticales) entre lesquelles peut se déplacer la raquette.
LR	Longueur de la raquette.
XM	Abscisse maximum de la raquette, compte tenu de sa longueur.
NB	Nombre total de briques.
M$	Caractère graphique utilisé pour représenter le cadre.
BR$	Caractère graphique représentant une brique.
BA$	Caractère graphique représentant la balle.
RA$	Chaîne de caractères graphiques représentant la raquette.
B$	Chaîne de caractères "espace" servant à effacer la raquette.

T	Compteur de temps.
ND	Nombre de briques détruites.
NC	Nombre de balles restant à jouer.
X,Y	Coordonnées de la balle.
DX,DY	Déplacements élémentaires de la balle (valent − 1 ou + 1).
XR,YR	Coordonnées de l'extrémité gauche de la raquette.
C1,C2,C3	Couleurs des caractères situés sur la trajectoire de la balle (voir schéma).
SC	Score.

Variables auxiliaires

R$ A

X,Y (dans le sous-programme 8000 seulement)

D

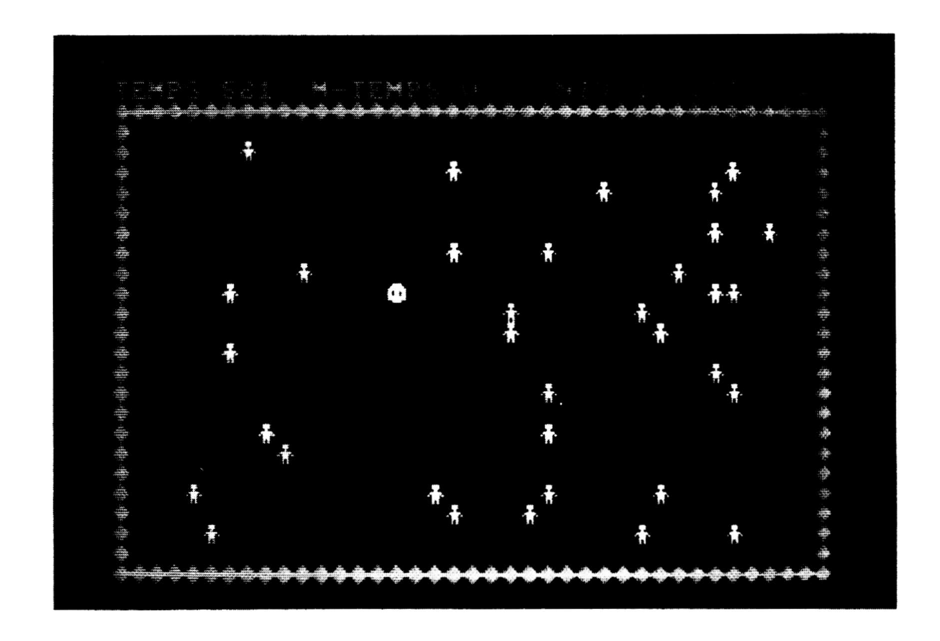
15 Embarquement immédiat

1. Le jeu

Devenez pour un instant un extra-terrestre ! C'est ce que vous propose ce jeu qui vous place aux commandes d'une soucoupe volante.

Vos compatriotes ont voulu réaliser une exploration pacifique de la terre. Ils y ont déposé une colonie de robots, programmés pour recueillir le maximum d'informations. Mais les choses ont mal tourné ! Les terriens se sont emparés de vos robots et tentent d'en percer le secret.

Vous êtes chargés de les récupérer le plus vite possible à l'aide d'une soucoupe équipée d'un générateur d'ondes anti-gravitationnelles. Vous pouvez ainsi « aspirer » n'importe quel objet, simplement en le survolant. Facile, direz-vous ! Certes. Encore, faut-il éviter de récupérer par maladresse des robots terriens qui voisinent avec les vôtres. En effet, ceux-ci risquent d'endommager plus ou moins



gravement votre soucoupe, vous obligeant ainsi à effectuer des réparations coûteuses en temps.

Le domaine où sont parqués vos robots est entouré d'un intense champ gravitationnel. Le moindre survol vous ferait vous écraser au sol.

Enfin, votre soucoupe consomme énormément d'énergie ; vous ne disposez que d'un temps limité pour effectuer votre « embarquement ». Si vous n'y parvenez pas dans le délai imparti, vous vous écraserez sur le sol terrestre.

Le programme vous propose quatre vitesses de jeu et trois niveaux :

- *Niveau 1 :* Destiné à l'entraînement. Il ne comporte ni robots terriens, ni champ gravitationnel.
- *Niveau 2 :* Il comporte des robots terriens, mais pas de champ gravitationnel.
- *Niveau 3 :* Il correspond à la description faite ci-dessus avec robots terriens et champ gravitationnel.

Vous voyez ensuite apparaître vos robots (en jaune), les robots terriens (en bleu) s'il y a lieu et votre soucoupe (en jaune). Le tout est entouré d'une bordure rose qui délimite le domaine de jeu et qui, au niveau 3, représente le champ antigravitationnel.

Vous pouvez examiner la situation à loisir et taper sur une touche lorsque vous êtes prêt à commencer. Vous dirigez votre soucoupe à l'aide des quatre touches

→ ← ↑ et ↓ (plus précisément, vous en modifiez la direction de déplacement car elle ne reste jamais immobile).

Le programme affiche en permanence le temps restant, le niveau de jeu, la vitesse choisie et le meilleur score déjà réalisé. Le survol d'un robot terrien est accompagné d'un bruit caractéristique et votre soucoupe est immobilisée pendant un certain temps (aléatoire). Le survol d'un de vos robots se traduit par ''zap''.

Si vous parvenez à atteindre votre objectif dans le délai imparti, vous entendrez votre soucoupe s'éloigner de plus en plus rapidement. Dans le cas contraire, vous l'entendrez tomber au sol.

2. Le programme

```
100 '*************** EMBARQUEMENT IMMEDIAT *******************
110 '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+11,16*(25-Y)+7)
130  GOSUB 9000
140  GOSUB 7000: GOSUB 8000
150  R$=INKEY$: IF R$<>"" THEN DL=ASC(R$)
160   XP=X+(DL=242)-(DL=243): YP=Y+(DL=240)-(DL=241)
170   T=T-1: LOCATE #1,6,1: PRINT #1,T;
180   IF VI<4 THEN FOR I=1 TO 25*(4-VI): NEXT I
190   IF FNSC(XP,YP)<>0 THEN 250
200   LOCATE X,Y: PRINT " ";
210   X=XP: Y=YP: LOCATE X,Y: PRINT S$;
220  IF T>0 AND NE>0 THEN 150
230  IF T<=0 THEN GOSUB 2000 ELSE GOSUB 3000
240  GOTO 310
250  C=FNSC(XP,YP)
260  IF C=3 AND NV=3 THEN GOSUB 2000:GOTO 310
270  IF C=3 THEN DL=0: GOTO 220
280  IF C=2 THEN GOSUB 1700: GOTO 220
290  IF C=1 THEN GOSUB 1500: GOTO 220
300  GOTO 150
310  IF T>MS(NV,VI) THEN MS(NV,VI)=T
320  LOCATE 3,25
330  WHILE INKEY$<>"": WEND
340  PRINT "voulez vous rejouer";
350  R$=INKEY$:IF R$="" THEN 350
360 IF R$<>"N" THEN 140
370 END
```

```
380
1500 '------------ SP EXTRA TERRESTRE EMBARQUE ---------------
1510 '
1520 FOR H=300 TO 20 STEP -10: SOUND 1,H,1: NEXT H
1530 NE=NE-1
1540 LOCATE X,Y: PRINT " ";
1550 X=XP: Y=YP: LOCATE X,Y: PRINT S$;
1560 RETURN
1570 '
1700 '------------ SP ROBOT TERRIEN TOUCHE -------------------
1710 '
1720 SOUND 1,2500,20
1730 LOCATE X,Y: PRINT " ";
1740 X=XP: Y=YP: LOCATE X,Y: PRINT S$;
1750 P=INT(RND(1)*60)
1760 FOR I=1 TO P
1770  SOUND 1,2000,1: SOUND 1,0,10
1780  T=T-1: LOCATE #1,6,1: PRINT #1,T;
1790 NEXT I
1800 RETURN
1810 '
2000 '------------ SP PERDU -----------------------------------
2010 '
2020 FOR K=1 TO 200: NEXT K
2030 T=0
2040 FOR H=10 TO 500: SOUND 1,H,1: NEXT H
2050 FOR K=1 TO 1000: NEXT K
2060 RETURN
2070 '
3000 '------------ SP GAGNE -----------------------------------
3010 '
3020 FOR K=1 TO 300: NEXT K
3030 FOR H=500 TO 100 STEP -1: SOUND 1,H,1: NEXT H
3040 FOR H=100 TO 10 STEP -1: SOUND 1,H,3: NEXT H
3050 RETURN
3060 '
7000 '------------ SP CHOIX NIVEAU ET VITESSE ----------------
7010 '
7020 CLS: PEN 3
7030 LOCATE 9,3: PRINT "EMBARQUEMENT";
7040 LOCATE 13,6: PRINT "IMMEDIAT";
7050 '
7060 LOCATE 3,9: PRINT "3 niveaux de jeu";
7070 LOCATE 5,10: PRINT "1 : pas d'obstacles";
```

```
7080 LOCATE 5,11: PRINT "2 : obstacles (robots terriens)";
7090 LOCATE 5,12: PRINT "3 : obstacles et champ de gravite";
7100 LOCATE 9,14: PRINT "niveau choisi?";
7110 R$=INKEY$: IF R$="" THEN 7110
7120 NV=VAL(R$)
7130 IF NV<=0 OR NV>3 THEN 7110
7140 PRINT NV;
7150 '
7160 LOCATE 3,16: PRINT "4 vitesses possibles";
7170 LOCATE 6,17: PRINT "de 1 (lent)";
7180 LOCATE 6,18: PRINT " a 4 (rapide)";
7190 LOCATE 9,20: PRINT "vitesse choisie?";
7200 R$=INKEY$: IF R$="" THEN 7200
7210 VI=VAL(R$)
7220 IF VI<=0 OR VI>4 THEN 7200
7230 PRINT VI;
7240 FOR K=1 TO 1000: NEXT K
7250 RETURN
7260 '
8000 '------------ SP INITIALISATION PARTIE-------------------
8010 '
8020 T=TX: NR=NRX: NE=NEX:
8030 CLS: CLS #1
8040 PEN 3
8050 FOR X=X1 TO X2
8060  LOCATE X,Y1: PRINT M$;
8070  LOCATE X,Y2: PRINT M$;
8080 NEXT X
8090 FOR Y=Y1 TO Y2
8100  LOCATE X1,Y: PRINT M$;
8110  LOCATE X2,Y: PRINT M$;
8120 NEXT Y
8130 '
8140 IF NV=1 THEN 8200
8150 PEN 2
8160 FOR I=1 TO NR
8170  GOSUB 8500: LOCATE X,Y: PRINT RB$;
8180 NEXT I
8190 '
8200 PEN 1
8210 FOR I=1 TO NE
8220  GOSUB 8500:LOCATE X,Y: PRINT ET$;
8230 NEXT I
8240 '
8250 GOSUB 8500: LOCATE X,Y: PRINT S$;
```

```
8260 '
8270 LOCATE #1,2,1
8280 WHILE INKEY$<>"": WEND
8290 PRINT #1, "tapez sur une touche pour commencer";
8300 R$=INKEY$: IF R$="" THEN 8300
8310 '
8320 LOCATE #1,1,1
8330 PRINT #1, "TEMPS       M-TEMPS       NIV    VIT   ";
8340 LOCATE #1,19,1: PRINT #1, MS(NV,VI);
8350 LOCATE #1,28,1: PRINT #1, NV;
8360 LOCATE #1,37,1: PRINT #1, VI;
8370 RETURN
8380 '
8500 '------------ SP TIRAGE COORDONNEES D'UN POINT ----------
8510 '
8520 X=X1+2+INT(RND(1)*(X2-X1-3))
8530 Y=Y1+2+INT(RND(1)*(Y2-Y1-3))
8540 IF FNSC(X,Y)<>0 THEN 8520
8550 RETURN
8560 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 BR=0: CF=0: CS=24: CR=23: CD=16
9030 X1=1: X2=39: Y1=2: Y2=25
9040 TX=800: NRX=15: NEX=25: DL=0
9050 '
9060 SYMBOL AFTER 160
9070 SYMBOL 160, 60,126,255,219,219,255,255,60
9080 SYMBOL 161, 28,28,8,62,93,28,20,20
9090 SYMBOL 162, 28,28,8,62,93,28,20,20
9100 SYMBOL 163, 24,60,126,255,255,126,60,24
9110 S$=CHR$(160): RB$=CHR$(161): ET$=CHR$(162): M$=CHR$(163)
9120 '
9130 INK 0,CF: INK 1,CS: INK 2,CR: INK 3,CD
9140 PAPER 0: CLS: BORDER BR
9150 WINDOW #1, 1,40,1,1: PAPER #1,0: PEN #1,3: CLS #1
9160 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier la couleur**

- *du fond :* remplacez, en 9020, la valeur 0 (dans CF = 0) par un nombre de votre choix.

- *des robots terriens :* remplacez, en 9020, la valeur 23 (dans CR = 23) par un nombre de votre choix.

- *de la soucoupe et de vos robots :* remplacez, en 9020, la valeur 24 (dans CS = 24) par un nombre de votre choix.

▷ **Pour changer la durée du jeu**

Remplacez, en 9040, la valeur 800 (dans TX = 800) par un nombre de votre choix.

▷ **Pour modifier le nombre :**

- *de robots terriens :* remplacez, en 9040, la valeur 15 (dans NRX = 15) par un nombre de votre choix.
- *de robots extra-terrestres :* remplacez, en 9040, la valeur 25 (dans NEX = 25) par un nombre de votre choix.

▷ **Pour utiliser une poignée de jeu**

Remplacez les lignes 150 et 160 par :

```
150  R=JOY(0) : IF R<>0 THEN DL=R
160  XP=X+(DL=4) - (DL=8) : YP=Y+(DL=1) - (DL=2)
```

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard,

— Animation,

— Lecture rapide du clavier,

— Examen du contenu de l'écran.

b) Le programme principal (100-370)

120 définit une fonction permettant de connaître le numéro d'encre d'*un* point graphique de l'emplacement texte de coordonnées X, Y.

130 appelle le sous-programme d'initialisation du jeu.

140-360 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme de choix du niveau et de la vitesse du jeu ainsi que le sous-programme d'initialisation d'une partie.

150-300 sont répétées jusqu'à ce que vous ayez embarqué tous les robots ou que le temps maximum soit dépassé ou que vous survoliez un champ gravitationnel :

150 examine le clavier et actualise, s'il y a lieu, la direction de déplacement de la soucoupe.

160 calcule la nouvelle position de la soucoupe.

170 actualise le temps.

180 réalise une attente dépendant de la vitesse de jeu choisie.

190 examine le *prochain emplacement* de la soucoupe :

200-230 *cas où il est disponible* :

200-210 déplacent la soucoupe.

220 examine si la partie est terminée (robots embarqués ou temps épuisé).

230 appelle, suivant le cas, l'un des deux sous-programmes de fin : ''perdu'' ou ''gagné''.

250-300 *cas où il est occupé :*

250-260 appellent le sous-programme de fin de partie s'il s'agit d'un champ gravitationnel et si l'on est en niveau 3.

270 immobilise la soucoupe s'il s'agit d'un champ gravitationnel et si le niveau de jeu est différent de 3.

280 appelle, s'il y a lieu, le sous-programme ''robot terrien touché''.

290 appelle, s'il y a lieu, le sous-programme ''robot embarqué''.

310-350 actualisent le meilleur score et vous demandent si vous souhaitez rejouer.

c) Le sous-programme d'initialisation du jeu (9000-9160)

9010 fixe les couleurs.

9030 fixe les limites du domaine du jeu.

9040 fixe la durée maximum d'une partie, le nombre de robots terriens et le nombre de robots extra-terrestres.

9060-9110 fabriquent les caractères graphiques représentant la soucoupe, les robots terriens, les robots extra-terrestres et le champ gravitationnel.

9130-9150 fixent les couleurs et définissent la fenêtre numéro 1 destinée à l'affichage des scores.

d) Le sous-programme de choix du niveau et de la vitesse de jeu (7000-7250)

7020-7040 affichent le titre du jeu.

7060-7140 font choisir le niveau de jeu, en rejetant les réponses autres que 1, 2 ou 3.

7160-7230 font choisir la vitesse de jeu, en rejetant les réponses autres que 1, 2, 3 ou 4.

7240 attend un court instant.

e) Le sous-programme d'initialisation d'une partie (8000-8370)

8020 initialise le compteur de temps, le nombre de robots terriens et le nombre de robots extra-terrestres.

8030-8120 dessinent le contour d'un rectangle représentant la zone où règne un champ gravitationnel.

8130-8180 dessinent les robots terriens, s'il y a lieu (niveau de jeu au moins égal à 2). Le sous-programme 8500 est utilisé pour tirer les coordonnées d'un emplacement disponible.

8190-8230 dessinent les robots extra-terrestres. Le sous-programme 8500 est utilisé pour tirer les coordonnées d'un emplacement disponible.

8240-8250 dessinent la soucoupe dans sa position initiale.

8270-8300 attendent que vous pressiez une touche.

8310-8370 affichent le meilleur score, le niveau et la vitesse de jeu.

f) Le sous-programme "robot extra-terrestre embarqué" (1500-1560)

1520 émet un son approprié.

1530 actualise le nombre de robots restant.

1540 efface le robot.

1550 déplace la soucoupe.

g) Le sous-programme "robot terrien touché" (1700-1800)

1720 émet un son approprié.

1730 efface le robot.

1740 déplace la soucoupe.

1750-1790 attendent une durée tirée au hasard et actualisent le compteur de temps.

5. Liste des variables

BR	Couleur du tour de l'écran.
CF	Couleur du fond (domaine de jeu).
CS	Couleur de la soucoupe et des robots extra-terrestres.
CR	Couleur des robots terriens.
CD	Couleur d'écriture des messages et du champ gravitationnel.
X1,X2,Y1,Y2	Coordonnées du domaine de jeu.
TX	Temps maximum d'une partie.
NRX	Nombre de robots terriens.
NEX	Nombre de robots extra-terrestres.
S$	Caractère graphique représentant la soucoupe.
RB$	Caractère graphique représentant les robots terriens.
ET$	Caractère graphique représentant les robots extra-terrestres.
M$	Caractère graphique utilisé pour représenter les limites du domaine.
NV	Niveau de jeu (de 1 à 3).
VI	Vitesse de jeu (de 1 à 4).
T	Compteur de temps.
NR	Nombre de robots terriens.
NE	Nombre de robots restant à embarquer.
X, Y	Coordonnées de la soucoupe.
XP, YP	Coordonnées du prochain emplacement de la soucoupe.
MS(3,4)	Tableau des meilleurs scores. MS(I,J) contient le meilleur score relatif au niveau I et à la vitesse J.

DL	Direction de déplacement de la soucoupe (240, 241, 242 ou 243).

Variables auxiliaires

R$ R A I

C C$

X,Y (dans les sous-programmes 8000 et 8500).

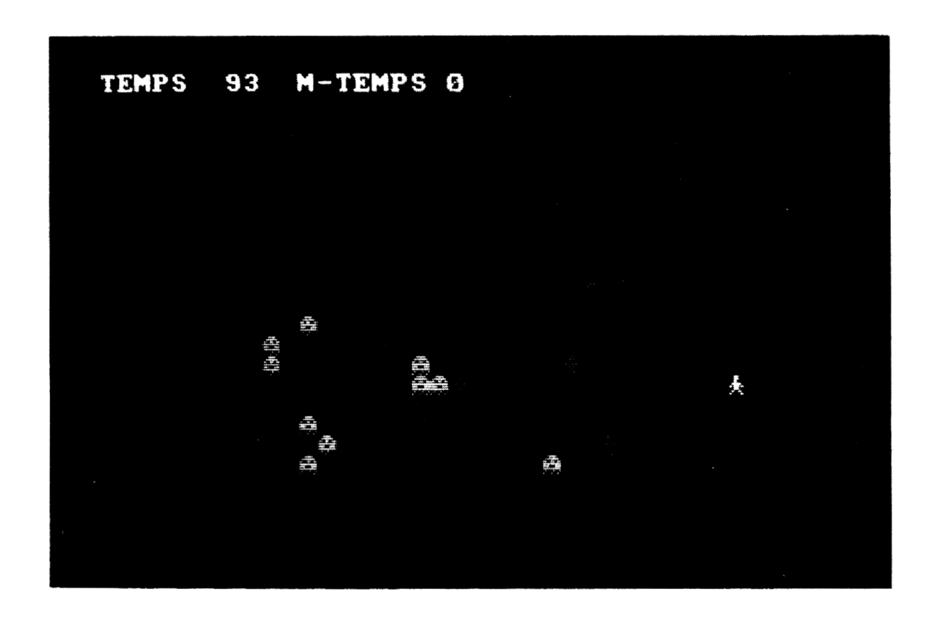
16
Aliénation

1. Le jeu

Çà y est ! Les aliens, ces êtres venus d'ailleurs, ont envahi la terre. La race humaine a presque entièrement disparu de la planète. Vous êtes l'un des rares survivants. Pour combien de temps ! De nombreux aliens ne cessent de vous poursuivre pour vous détruire. Or, le moindre contact avec l'une de ces créatures serait mortel.

Pour comble de tout, vous ne disposez d'aucune arme pour vous défendre. Toutefois, ça et là gisent des robots, à forme humaine, cloués sur place par une privation soudaine de l'énergie nécessaire à leur fonctionnement. Que puis-je attendre d'eux me direz-vous ?

En fait, chacun de ces robots va pouvoir vous rendre un ultime service en faisant disparaître un alien (et un seul). En effet, tout contact entre un alien et un objet métallique conduit à la désintégration complète, et de l'objet et de l'alien. D'autre



part, les aliens sont des êtres un peu stupides et aux réflexes lents (c'est leur nombre seul qui fait leur force).

Ils se déplacent par bonds irréguliers et, de façon assez grossière, dans votre direction. Il vous suffit donc de les obliger à bondir sur un robot pour les faire disparaître.

Comme vous le constaterez, les aliens font de grands bonds lorsqu'ils sont loin de vous. Par contre, une sorte de méfiance instinctive les amène à avancer plus lentement lorsqu'ils sont plus proches de vous. D'autre part, leur sens de l'orientation est peu évolué puisqu'ils peuvent être tout près de vous et faire un bond sur... "un emplacement voisin".

Enfin, sachez que vous êtes la seule proie qui intéresse les aliens. En conséquence, ils ne voient aucunement les robots et leurs déplacements ne sont nullement affectés par leur présence.

En début de partie, vous voyez apparaître la disposition des aliens (violets), des robots (bleus) et du personnage qui vous représente (en jaune). Vous pouvez prendre le temps d'étudier la situation avant d'appuyer sur une touche quelconque. La chasse infernale s'engage alors. Vous vous déplacez grâce aux quatre touches fléchées ? Ne craignez pas de percuter un robot. Cela ne vous fera aucun mal.

Chaque désintégration (robot + alien) vous est signalée par un bruit sec. Si vous êtes atteint par un alien (ou si, maladroitement, vous en percutez un) votre disparition est accompagnée de sons et d'effets lumineux appropriés. Le programme affiche en permanence le temps qui s'écoule et le meilleur temps réalisé jusqu'ici.

2. Le programme

```
100 '******************** ALIENATION ************************
110 '
120 DEF FNSC(X,Y) = TEST(16*(X-1)+7,16*(25-Y)+13)
130 GOSUB 9000
140  GOSUB 8000
150  FOR I=1 TO NA
160   T=T+1: LOCATE #1, 8,1: PRINT #1,T;
170   GOSUB 4000: GOSUB 5000: IF FI=1 THEN 200
180  NEXT I
190  GOTO 150
200  IF CT=0 AND (T<MT OR MT=0) THEN MT=T
210  FOR K=1 TO 500: NEXT K
220  WHILE INKEY$<>"": WEND
230  LOCATE #1,26,1: PRINT #1,"rejouez vous";
240  R$=INKEY$: IF R$="" THEN 240
250 IF R$<>"N" THEN 140
260 END
270 '
4000 '------------ SP DEPLACEMENT HUMANOIDE -----------------
4010 '
4020 DG=INKEY(8): DD=INKEY(1): DH=INKEY(0): DB=INKEY(2)
4030 IF DG*DD*DH*DB<>0 THEN RETURN
4040 XN=XB+(DG=0)-(DD=0): YN=YB+(DH=0)-(DB=0)
4050 IF XN>X2-2 OR XN<X1+2 OR YN>Y2-2 OR YN<Y1+2 THEN RETURN
4060 C=FNSC(XN,YN)
4070 IF C=3 THEN GOSUB 6000: RETURN
4080 IF C=2 THEN RETURN
4090 PEN 1: LOCATE XB,YB: PRINT " ";
4100 XB=XN: YB=YN: LOCATE XB,YB: PRINT H$;
4110 RETURN
4120 '
5000 '------------ SP DEPLACEMENT ALIEN NUMERO I ------------
5010 '
5020 X=XA(I): Y=YA(I): IF X=0 THEN RETURN
5030 XN=X+(RND(1)+0.5)*((X>XB)-(X<XB)+2*((X>XB+4)-(X<XB-4)))
5040 YN=Y+(RND(1)+0.5)*((Y>YB)-(Y<YB)+2*((Y>YB+4)-(Y<YB-4)))
5050 XN=INT(XN): YN=INT(YN)
5060 C=FNSC(XN,YN): IF C<>0 THEN 5110
5070  PEN 3: LOCATE X,Y: PRINT " ";
5080  LOCATE XN,YN: PRINT A$;
5090  XA(I)=XN: YA(I)=YN
5100  RETURN
```

```
5110 IF C=3 THEN RETURN
5120 IF C=2 THEN GOSUB 7000
5130 IF C=1 THEN GOSUB 6000
5140 RETURN
5150 '
6000 '------------ SP MORT HUMANOIDE -------------------------
6010 '
6020 FOR K= 0 TO 26
6030  INK 0,K
6040  FOR H=300 TO 100 STEP -20: SOUND 1,H,1: NEXT H
6050 NEXT K
6060 INK 0,0
6070 LOCATE XB,YB: PRINT " ";
6080 PEN 3: LOCATE XN,YN: PRINT A$;
6090 FI=1
6100 RETURN
6110 '
7000 '------------ SP MORT ALIEN ------------------------------
7010 '
7020 LOCATE X,Y: PRINT " ";: XA(I)=0
7030 PEN 2: LOCATE XN,YN: PRINT A$;
7040 FOR H=300 TO 500 STEP 9: SOUND 1,H,1: NEXT H
7050 LOCATE XN,YN: PRINT " ";
7060 CT=CT-1: IF CT<=0 THEN FI=1
7070 RETURN
7080 '
8000 '------------ SP INITIALISATION PARTIE ------------------
8010 '
8020 CLS: CLS #1
8030 CT=NA: T=0: FI=0
8040 '
8050 PEN 2
8060 FOR I=1 TO NR
8070  X=X1+INT(RND(1)*(X2-X1-5))
8080  Y=Y1+INT(RND(1)*(Y2-Y1-5))
8090  IF FNSC(X,Y)<>0 THEN 8070
8100  LOCATE X,Y: PRINT RB$;
8110 NEXT I
8120 '
8130 PEN 3
8140 FOR I=1 TO NA
8150  X=X1+INT(RND(1)*(X2-X1+1))
8160  Y=Y1+INT(RND(1)*(Y2-Y1+1))
8170  IF FNSC(X,Y)<>0 THEN 8160
```

```
8180  XA(I)=X: YA(I)=Y
8190  LOCATE X,Y: PRINT A$;
8200 NEXT I
8210 '
8220 XB=X1+INT(RND(1)*(X2-X1-4))+2
8230 YB=Y1+INT(RND(1)*(Y2-Y1-4))+2
8240 IF FNSC(XB,YB)<>0 THEN 8220
8250 PEN 1: LOCATE XB,YB: PRINT H$;
8260 '
8270 LOCATE #1,1,1
8280 PRINT #1,"pour commencer, tapez surune touche";
8290 R$=INKEY$: IF R$="" THEN 8290
8300 CLS #1
8310 LOCATE #1,2,1: PRINT #1,"TEMPS      M-TEMPS"; MT;
8320 RETURN
8330 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 CE=0: BR=0: CH=24: CR=10: CA=7
9030 X1=3: X2=39: Y1=3: Y2=25
9040 NR=18: NA=12
9050 DIM XA(NA), YA(NA)
9060 '
9070 SYMBOL AFTER 160
9080 SYMBOL 160, 24,24,16,60,90,24,36,66
9090 SYMBOL 161, 28,28,8,127,28,28,20,20
9100 SYMBOL 162, 60,126,219,255,231,255,73,146
9110 H$=CHR$(160): RB$=CHR$(161): A$=CHR$(162)
9120 '
9130 MODE 1
9140 INK 0,CE: INK 1,CH: INK 2,CR: INK 3,CA
9150 PAPER 0: CLS: BORDER BR
9160 WINDOW #1, 1,40,1,1: PAPER #1,0: PEN #1,1: CLS#1
9170 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ **Pour modifier le nombre de robots**

Remplacez, en 9040, le nombre 18 (dans NR = 18) par la valeur de votre choix.

▷ **Pour modifier le nombre d'aliens**

Remplacez, en 9040, le nombre 12 (dans NA = 12) par la valeur de votre choix.

REMARQUE : ne placez jamais plus d'aliens que de robots, car il serait alors impossible de gagner.

▷ **Pour utiliser une poignée de jeu**

```
4020 RJ=JOY(0) : DG=RJ AND 8 : DD=RJ AND 4
4030 DH=RJ AND 1 : DB=RJ AND 2 : IF DG+DD+DB+DH=0 THEN RETURN
4040 XN=XB-(DG=8)+(DD=4) : YN=YB-(DB=2)+(DH=1)
```

4. Description du programme

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Lecture rapide du clavier,
- Examen du contenu de l'écran.

b) Le programme principal (100-260)

120 définit une fonction permettant de connaître la couleur d'encre d'*un* point graphique de l'emplacement texte de coordonnées X, Y.

130 appelle le sous-programme d'initialisation du jeu.

140-250 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150-190 sont répétées jusqu'à ce que la partie soit finie (FI=1), soit par suppression de tous les aliens, soit par votre propre disparition :

150-180 répètent, pour chaque alien, l'incrémentation du compteur de temps et l'appel des sous-programmes ''déplacement humanoïde'' et ''déplacement d'un alien''.

200 actualise le meilleur score.

210-240 vous demandent si vous voulez rejouer.

250 examine votre réponse.

c) Le sous-programme d'initialisation du jeu (9000-9170)

9020 détermine les couleurs.

9030 fixe les limites du domaine de jeu.

9040 fixe le nombre de robots et d'aliens.

9050 réserve les deux tableaux contenant les coordonnées des aliens.

9070-9110 fabriquent les caractères graphiques représentant l'humanoïde, les robots et les aliens.

9130-9160 fixent les couleurs et définissent la fenêtre numéro 1 destinée à l'affichage des scores.

d) Le sous-programme d'initialisation d'une partie (8000-8320)

8020 efface l'écran.

8030 initialise le compteur d'aliens survivants, le compteur de temps et l'indicateur de fin de partie.

8040-8110 dessinent les robots.

8120-8200 dessinent les aliens dans leur position et placent leurs coordonnées dans les tableaux XA et YA.

8210-8250 dessinent l'humanoïde dans sa position initiale.

8260-8290 attendent que vous pressiez une touche.

8310 affiche le meilleur temps.

e) Le sous-programme de déplacement d'un alien (5000-5140)

Il déplace l'alien dont le numéro est compris dans la variable I.

5020 vérifie que l'alien concerné est encore "vivant".

5020-5050 calculent la nouvelle position de l'alien.

5060 examine le contenu de cette nouvelle position.

5070-5100 correspondent au cas où elle est disponible ; elles déplacent l'alien.

5110-5130 correspondent au cas où elle est occupée :

- 5110 examine s'il s'agit d'un autre alien.
- 5120 appelle, s'il s'agit d'un robot, le sous-programme "mort d'un alien".
- 5130 appelle, s'il s'agit de l'humanoïde, le sous-programme "mort de l'humanoïde".

f) Le sous-programme de déplacement de l'humanoïde (4000-4110)

4020-4030 examinent le clavier.

4040-4050 calculent les nouvelles coordonnées de l'humanoïde et vérifient qu'elles restent à l'intérieur du domaine de jeu.

4060 détermine le contenu de la nouvelle position :

4070 appelle le sous-programme ''mort de l'humanoïde'', s'il s'agit d'un alien.

4080 examine s'il s'agit d'un robot.

4090-4100 déplacent l'humanoïde lorsque la nouvelle position est disponible.

g) Le sous-programme ''mort de l'humanoïde'' (6000-6100)

6020-6050 créent des effets ''sonores et lumineux''.

6060 rétablit la couleur de fond d'origine.

6070-6080 font disparaître l'humanoïde sous l'alien.

6090 positionne l'indicateur de fin de partie.

h) Le sous-programme ''mort d'un alien'' (7000-7070)

7020-7050 déplacent provisoirement l'alien à l'emplacement occupé par un robot, émettent un son approprié, puis le font disparaître.

7060 décrémente le compteur d'aliens survivants ; positionne à un l'indicateur de fin de partie lorsque tous les aliens ont disparu.

5. Liste des variables

CE	Couleur de fond du domaine de jeu.
BR	Couleur du tour de l'écran.
CH	Couleur de l'humanoïde et des scores.
CR	Couleur des robots.
CA	Couleur des aliens.
X1,X2,Y1,Y2	Coordonnées du domaine de jeu.
NR	Nombre total de robots.
NA	Nombre total d'aliens.

XA(NA),YA(NA)	Tableaux contenant les coordonnées des aliens. Quand un alien disparaît, son abscisse prend la valeur zéro.
H$	Caractère graphique représentant l'humanoïde.
RB$	Caractère graphique représentant les robots.
A$	Caractère graphique représentant les aliens.
B$	Chaîne de 36 caractères ''espace'' utilisée pour l'effacement de ''messages''.
CT	Nombre d'aliens survivants.
T	Temps écoulé.
FI	Indicateur de fin de partie : FI = 0 cas usuel FI = 1 fin de partie détectée.
XB,YB	Coordonnées de l'humanoïde.
XN,YN	Nouvelles coordonnées de l'humanoïde ou d'un alien.
I	Numéro de l'alien en cours de déplacement.
C	Contenu d'une case.
MT	Meilleur temps.

Variables auxiliaires

R$	A
I	(dans le sous-programme 8000 seulement)
X	Y

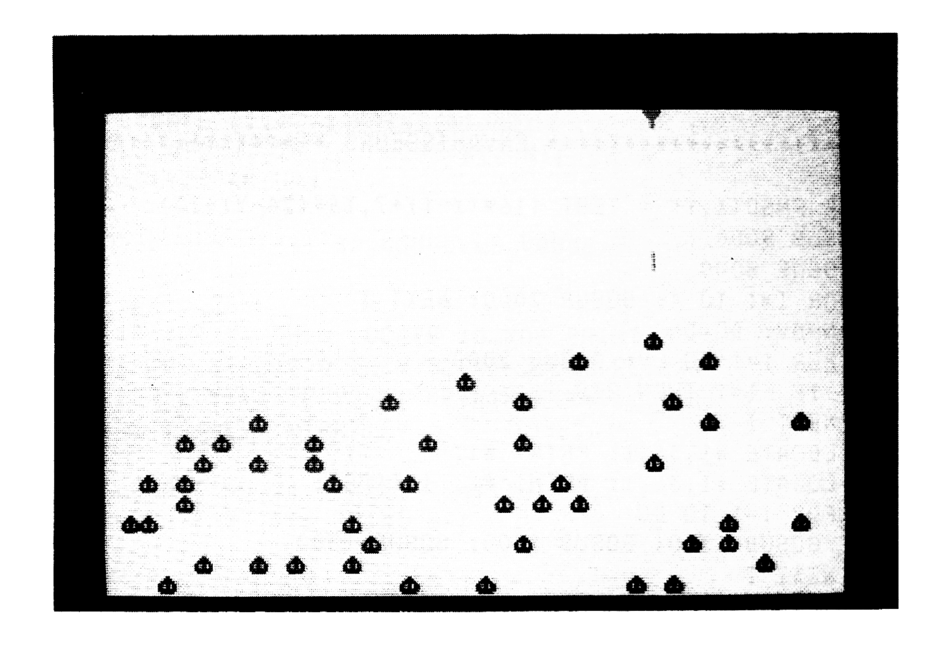
17
Les envahisseurs

1. Le jeu

Les extra-terrestres envahissent notre planète. Bien que non armés, leurs intentions ne sont nullement pacifiques. Vous êtes chargés d'en détruire le maximum à l'aide de missiles classiques.

Les envahisseurs se présentent par vagues de dix soucoupes. Leur entrée brutale dans l'atmosphère terrestre entraîne des perturbations de tout genre dont les deux plus importantes sont : violente illumination du ciel, neutralisation de votre base de tir (elle ne peut ni se déplacer, ni tirer de missiles).

Les envahisseurs ont prévu que les différentes vagues se succèdent de plus en plus rapidement. Cependant, ils tiennent compte de votre habileté à les détruire. Plus vous serez rapides, plus le temps séparant deux vagues sera élevé. Votre ordinateur central, couplé à vos radars vous renseignera sur le délai prévu entre



une vague et la suivante. Vous pourrez ainsi mesurer l'effet de votre effort destructeur.

Au début du jeu, trois vagues apparaissent successivement en bas de l'écran. Les soucoupes s'immobilisent alors dans l'attente de la vague suivante.

Vous voyez votre base de tir en haut de l'écran. Elle se déplace à l'aide des touches "←" et "→" et vous tirez un missile en pressant la barre d'espace. Vous ne pouvez tirer de nouveau missile tant que le précédent n'a pas atteint son but ou disparu en bas de l'écran. Par contre, vous pouvez déplacer votre base pendant qu'un missile est en l'air. Très souvent même, vous serez obligés de le faire pour gagner du temps.

Quand une nouvelle vague de dix soucoupes se présente, le ciel devient blanc et vous ne pouvez plus agir sur votre base de tir. Les soucoupes déjà présentes montent légèrement pour faire place aux nouveaux arrivants puis tout redevient normal (si l'on peut dire).

Le jeu se poursuit jusqu'à ce que l'une des soucoupes atteigne le niveau de votre base. Vous avez donc tout intérêt à éliminer les soucoupes les plus hautes.

Le programme affiche en permanence votre score (nombre de soucoupes détruites), le meilleur score, le nombre d'envahisseurs présents dans le ciel (N-EN) et le délai d'apparition de la vague suivante (DELAI).

2. Le programme

```
100 '******************** ENVAHISSEURS ***********************
110 '
120 DEF FNSC(X,Y) = TEST (16*(X-1)+7,16*(24-Y)+13)
130 GOSUB 9000
140  GOSUB 8000
150  FOR I=1 TO 4: GOSUB 2000: NEXT I
160  DN=DV: DC=DN
170   FOR I=1 TO LV: GOSUB 2000
180    IF FI=1 THEN 270
190   NEXT I
200   LOCATE #1,36,1: PRINT #1, "    ";
210   LOCATE #1,36,1: PRINT #1, INT(DC);
220   FOR I=1 TO DC
230    GOSUB 1500: GOSUB 1000: GOSUB 1500
240   NEXT I
250   DN=DN*ZT: DC=DN*(EV*30-NT)/(EV*30)
260  GOTO 170
270  IF SC>MS THEN MS=SC
280  FOR K=1 TO 500: NEXT K
290  WHILE INKEY$<>"": WEND
300  LOCATE 10,10
310  PRINT "voulez vous rejouer";
320  R$=INKEY$: IF R$="" THEN 320
330 IF R$<>"N" THEN 140
340 END
350 '
1000 '------------ SP DEPLACEMENT BASE DE TIR ---------------
1010 '
1020 IF INKEY(47)<>0 OR YO<>YT THEN 1040
1030  XO=XT: YO=YT+1: LOCATE XO,YO: PRINT M$;: RETURN
1040 DG=INKEY(8): DD=INKEY(1): IF DD*DG<>0 THEN RETURN
1050 XN=XT+(DG=0)-(DD=0): IF XN>X2 OR XN<X1 THEN RETURN
1060 LOCATE XT,YT: PRINT " ";: LOCATE XN,YT:PRINT B$;:XT=XN
1070 RETURN
1080 '
1500 '------------ SP DEPLACEMENT MISSILE -------------------
1510 '
1520 IF YO=YT THEN FOR K=1 TO 20:NEXT K: RETURN
1530 YN=YO+1: LOCATE XO,YO: PRINT " ";
1540 IF YN>Y2 THEN YO=YT: RETURN
1550 IF FNSC(XO,YN)=0THEN LOCATE XO,YN:PRINT M$;:YO=YN: RETURN
```

```
1560 FOR H=200 TO 120 STEP -10:SOUND 1,H,1: NEXT H
1570 LOCATE XO,YN: PRINT " ";
1580 NE(YN)=NE(YN)-1: SC=SC+1: NT=NT-1
1590 LOCATE #1,4,1; PRINT #1,SC;
1600 LOCATE #1,26,1: PRINT #1, NT;
1610 YO=YT: RETURN
1620 '
2000 '------------ SP NOUVELLE VAGUE ------------------------
2010 '
2020 INK 0,26
2030 IF YO<>YT THEN LOCATE XO,YO: PRINT " ";: YO=YT
2040 IF NE(YT+2)<>0 THEN GOSUB 3000: RETURN
2050 FOR K=YT+1 TO Y2-1: NE(K)=NE(K+1): NEXT K
2060 NE(Y2)=EV: NT=NT+EV
2070 '
2080 LOCATE 40,24: PRINT CHR$(10)
2090 LOCATE XT,YT: PRINT B$;
2100 PEN 2
2110 FOR K=1 TO EV
2120  X=INT(RND(1)*(X2-X1)+X1)
2130  IF FNSC(X,Y2)<>0 THEN 2120
2140  LOCATE X,Y2: PRINT S$;
2150 NEXT K
2160 PEN 1
2170 INK 0,CE
2180 LOCATE #1,26,1: PRINT #1,NT;
2190 RETURN
2200 '
3000 '------------ SP FIN DE PARTIE -------------------------
3010 '
3020 FOR K=0 TO 26
3030  INK 0,K
3040  FOR H=200 TO 100 STEP -5: SOUND 1,H,1: NEXT H
3050 NEXT K
3060 FOR K=1 TO 500 : NEXT K
3070 FI=1: INK 0,CE
3080 RETURN
3090 '
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 CLS: CLS #1
8030 LOCATE #1,2,1: PRINT #1, "SC       M-SC"; MS;
8040 LOCATE #1,22,1: PRINT #1, "N-EN      DELAI";
8050 '
```

```
8060 PEN 1
8070 XT=21: YT=1: XO=XT: YO=YT
8080 LOCATE XT,YT: PRINT B$;
8090 DE=DV: SC=0:FI=0: NT=0
8100 FOR I=1 TO Y2: NE(I)=0: NEXT I
8110 RETURN
8120 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 CE=23: BR=0: CT=2: CS=6: CF=0
9030 X1=2: X2=39: Y1=1: Y2=24
9040 DV=180: EV=5: LV=2: ZT=0.975
9050 MS=0: DIM NE(25)
9060 '
9070 SYMBOL AFTER 160
9080 SYMBOL 160, 255,255,126,60,24,24,24,24
9090 SYMBOL 161, 24,24,24,24,24,24,24,24
9100 SYMBOL 162, 24,60,126,219,219,255,126,0
9110 B$=CHR$(160): M$=CHR$(161): S$=CHR$(162)
9120 '
9130 MODE 1
9140 INK 0,CE: INK 1,CT: INK 2,CS: INK 3,CF
9150 WINDOW #1,1,40,1,1: PAPER #1,2: PEN #1,3: CLS #1
9160 WINDOW #0, 1,40,2,25: PAPER 0: PEN 1: CLS
9170 BORDER BR
9180 RETURN
```

3. Si vous souhaitez personnaliser ce programme

▷ Pour modifier la couleur du ciel

Remplacez, en 9020, la valeur 23 (dans CE = 23) par un nombre de votre choix compris entre 0 et 26.

▷ Pour modifier la couleur des soucoupes

Remplacez, en 9020, la valeur 6 (dans CS = 6) par un nombre de votre choix compris entre 0 et 26.

▷ Pour modifier le nombre de soucoupes par vague

Vous pouvez agir à la fois sur le nombre de soucoupes par ligne (actuellement 5) et sur le nombre de lignes par vague (actuellement 2).

Pour modifier le nombre de soucoupes par ligne, remplacez, en 9040, la valeur 5 (dans EV = 5) par le nombre de votre choix.

Pour modifier le nombre de lignes par vague, remplacez, en 9040, la valeur 2 (dans LV = 2) par un nombre de votre choix.

▷ **Pour modifier le rythme de jeu** et le rendre ainsi plus facile ou plus difficile

Vous pouvez agir à la fois sur le délai initial séparant deux vagues (actuellement 180) et sur le coefficient réducteur de délai (actuellement 0.975).

Pour modifier le délai initial séparant deux vagues, remplacez, en 9040, la valeur 180 (dans DV = 180) par un nombre de votre choix.

Pour modifier le coefficient réducteur, remplacez, en 9040, la valeur 0.975 (dans ZT = 0.975) par un nombre de votre choix ne dépassant pas 1. Un nombre supérieur à 0.975 correspondra à une partie dont le rythme s'accélèrera moins rapidement. Un nombre plus petit correspondra à une accélération plus rapide du rythme.

Notez que si vous souhaitez que le délai séparant deux vagues soit constant, il vous suffit de choisir 1 comme coefficient.

▷ **Pour utiliser une poignée de jeu**

Remplacez la ligne 1020 par :

```
1020 RJ=JOY (0) : IF (RJ AND 16) <> 16 OR YO <> YT THEN 1040
```

et les lignes 1040 et 1050 par :

```
1040 DG=RJ AND 4 : DD=RJ AND 8 : IF DD+DG=0 THEN RETURN
1050 XN=XT+(DG=4)-(DD=8) : IF XN>X2 OR XN<X1 THEN RETURN
```

4. Description du programme

a) Techniques employées, décrites en annexe

- Hasard,
- Animation,
- Lecteur rapide du clavier,
- Examen du contenu de l'écran.

b) Le programme principal (100-320)

120 définit une fonction permettant de connaître le numéro d'encre d'*un* point graphique de l'emplacement texte de coordonnées X,Y.

130 appelle le sous-programme d'initialisation du jeu.

140-330 correspondent au déroulement d'une partie. Elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

140 appelle le sous-programme d'initialisation d'une partie.

150 fait apparaître quatre rangées d'envahisseurs.

160 initialise les valeurs des délais entre deux vagues (voir variables DC et et DN).

170-260 sont répétées jusqu'à ce que les envahisseurs aient atteint votre base (F1 = 1) :

170-190 font apparaître une nouvelle vague de soucoupes, en vérifiant qu'elle n'atteint pas votre base.

200-210 affichent le délai d'apparition de la prochaine vague.

220-240 répètent (un nombre de fois correspondant au délai) l'appel des sous-programmes de déplacement du missile et de la base de tir.

250 actualise les valeurs des délais.

270 actualise le meilleur score.

280-320 vous demandent si vous souhaitez rejouer.

330 examine votre réponse.

c) Le sous-programme d'initialisation du jeu (9000-9180)

9020 détermine les couleurs.

9030 fixe les limites du domaine de jeu.

9040 initialise le délai séparant deux vagues, fixe le nombre d'envahisseurs par ligne, le nombre de lignes par vague et le coefficient de réduction de délai.

9050 initialise le meilleur score et réserve l'emplacement du tableau servant à comptabiliser le nombre d'envahisseurs présents sur chaque ligne d'écran.

9060-9110 fabriquent les caractères graphiques représentant la base de tir, le missile et les soucoupes.

9130-9170 définissent les "fenêtres" et fixent les couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8110)

8020-8040 effacent l'écran et affichent les titres.

8070 initialise les positions de la base de tir et du missile.

8080 dessine la base de tir.

8090 initialise le délai séparant deux vagues, le score, l'indicateur de fin de partie et le nombre de soucoupes présentes sur l'écran.

8100 initialise le tableau : nombre de soucoupes par ligne d'écran.

e) Le sous-programme de déplacement de la base de tir (1000-1070)

1020 examine le clavier.

1030 initialise un tir si la barre d'espace a été pressée et si aucun autre tir n'est en cours.

1040-1050 calculent la nouvelle position de la base de tir, en vérifiant qu'elle ne sort pas des limites permises.

1060 déplace la base de tir.

f) Le sous-programme de déplacement du missile (1500-1610)

1520 examine si un tir est en cours ;

1530 calcule la nouvelle position du missile et l'efface de son ancienne position.

1540 annule le "tir en cours" si le missile sort de l'écran.

1550 déplace le missile si la nouvelle position est disponible.

1560-1610 correspondent au cas où une soucoupe est atteinte :

1560 émet un son approprié et efface la soucoupe.

1570-1600 actualisent le nombre de soucoupes par ligne, le score, le nombre de soucoupes présentes sur l'écran.

1610 annule le "tir en cours".

g) Le sous-programme "nouvelle vague d'envahisseurs" (2000-2190)

2020 donne au ciel la couleur "blanc".

2030 annule le tir en cours, s'il y a lieu (en effaçant le missile).

2040 vérifie que la "dernière" ligne ne contient pas d'envahisseurs. Si ce n'est pas le cas, le sous-programme "fin de partie" est appelé.

2050-2060 actualisent le nombre d'envahisseurs par ligne et le nombre total d'envahisseurs présents sur l'écran.

2080-2090 font défiler le contenu de l'écran vers le haut (scroll) et redessinent la base de tir.

2100-2150 placent au hasard cinq nouveaux envahisseurs en bas de l'écran.

2170 rétablit la couleur ''normale'' du ciel.

h) Le sous-programme de fin de partie (3000-3080)

3020-3050 font passer le ciel par ''toutes les couleurs'' en émettant des sons appropriés.

3060 attend un instant.

3070 positionne l'indicateur de fin de partie et redonne au ciel sa couleur ''normale''.

5. Liste des variables

CE	Couleur du ciel.
BR	Couleur du tour de l'écran.
CT	Couleur de la base de tir et des scores.
CS	Couleur des soucoupes.
CF	Couleur de fond du bandeau supérieur où s'affichent les scores.
X1,X2,Y1,Y2	Coordonnées du domaine de jeu.
DV	Délai maximum séparant deux vagues.
EV	Nombre d'envahisseurs par ligne.
LV	Nombre de lignes formant une vague.
ZT	Coefficient réducteur du délai séparant deux vagues.
NE(24)	Tableau contenant le nombre de soucoupes de chaque ligne d'écran.
MS	Meilleur score.
B$	Caractère graphique représentant la base de tir.
M$	Caractère graphique représentant le missile.
S$	Caractère graphique représentant les soucoupes.
XT,YT	Coordonnées de la base de tir.
XO,YO	Coordonnées du missile. Si YO = YT, aucun tir n'est en cours.
DC	Délai séparant deux vagues.

SC	Score.
FI	Indicateur de fin de partie : 0 : cas normal 1 : fin de partie
NT	Nombre total d'envahisseurs présents sur l'écran.
XN	Nouvelle abscisse de la base de tir.
YN	Nouvelle ordonnée du missile.

Variables auxiliaires

R$ A I DL

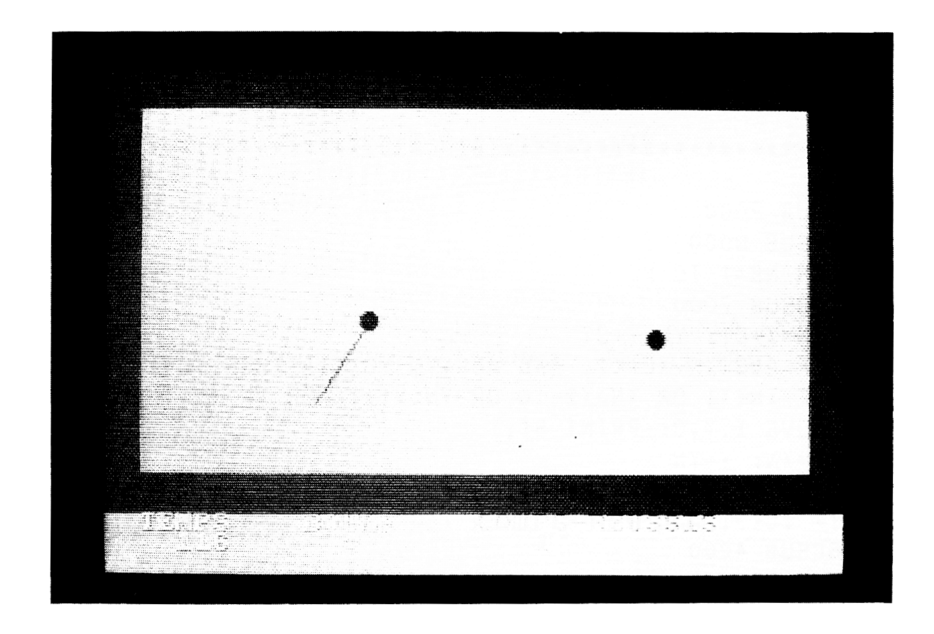
18 Billard

1. Le jeu

Vous allez pouvoir vous adonner aux joies de ce jeu, sans risquer de déchirer le tapis d'un coup de canne malencontreux.

Le but consiste à lancer votre boule, grâce à une canne, de manière à ce qu'elle vienne toucher une autre boule après deux rebonds sur les bords du billard. Si vous touchez sans rebond (ce qui est très facile) ou après un seul rebond, vous ne marquez aucun point.

Le programme commence par vous dessiner un tapis de billard (jaune) entouré d'une bordure violette. Vous voyez apparaître deux boules, en des emplacements quelconques. L'une d'entre elles est accompagnée d'une ''canne de billard'' représentée par un trait. Vous allez tout d'abord devoir mettre en place la canne dans la position qui vous semble la meilleure pour l'objectif visé (toucher l'autre boule



en deux rebonds). Vous utilisez pour cela les touches fléchées qui vous donnent deux vitesses de déplacement de la canne (autour de la boule) :

← : déplacement lent dans le sens inverse des aiguilles d'une montre.
↓ : déplacement rapide dans le sens des aiguilles d'une montre.
→ : déplacement lent dans le sens des aiguilles d'une montre.
↓ : déplacement rapide dans le sens des aiguilles d'une montre.

Quand la position vous paraît correcte, vous ''tirez'' en pressant la barre d'espace. Vous entendez alors le son de la canne heurtant votre boule puis celle-ci se déplace sur le tapis. Chaque rebond se signale par un son approprié.

Si vous touchez la boule dans les conditions requises vous serez récompensés par le jeu d'un accord.

Douze coups vous seront ainsi proposés. A la fin vous obtiendrez un score qui dépendra du nombre de coups réussis et du temps passé à mettre votre canne en place.

Le programme affiche en permanence le temps qui s'écoule pendant la mise en place de la canne, le numéro du coup et le nombre de coups réussis.

2. Le programme

```
100 '********************** BILLARD *************************
110 '
120 GOSUB 9000
130  GOSUB 8000
140  FOR NB=1 TO NC
150   GOSUB 7000
160   XN=X+CT: YN=Y+ST
170   IF ABS(XN-XC)<16 AND ABS(YN-YC)<16 THEN TC=1
180   IF XN<X1 OR XN>X2-16 OR YN<Y1+16 OR YN>Y2 THEN GOSUB 2000
190   MOVE X,Y,0: PRINT B$;
200   X=XN: Y=YN
210   MOVE X,Y,2: PRINT B$;
220  IF TC<>1 AND NR<=RX THEN 160
230  GOSUB 4000
240 NEXT NB
250 GOSUB 8500
260 IF R$<>"N" THEN 130
270 END
280 '
2000 '------------ SP DEPLACEMENT AVEC REBOND-----------------
2010 '
2020 IF XN<X1 THEN XN=X1: NR=NR+1: CT=-CT
2030 IF XN>X2-16 THEN XN=X2-16: NR=NR+1: CT=-CT
2040 IF YN<Y1+16 THEN YN=Y1+16: NR=NR+1: ST=-ST
2050 IF YN>Y2 THEN YN=Y2: NR=NR+1: ST=-ST
2060 SOUND 1,319,5
2070 RETURN
2080 '
4000 '------------ SP FIN D'UN COUP --------------------------
4010 '
4020 IF TC=1 THEN SOUND 1,478,10
4030 IF TC<>1 OR NR<>RX THEN 4070
4040  SOUND 1,478,20: SOUND 1,379,20
4050  SOUND 1,319,20: SOUND 1,239,20
4060  NT=NT+1
4070 LOCATE #1,23,2: PRINT #1, NT;
4080 FOR K=1 TO 500: NEXT K
4090 MOVE XC,YC,0: PRINT B$;
4100 MOVE X,Y,0: PRINT B$;
4110 RETURN
4120 '
```

```
6000 '------------ SP MISE EN PLACE CANNE --------------------
6010 '
6020 T=T+1: GOSUB 6500
6030 XE=X+8+LC*COS(TH): YE=Y-7+LC*SIN(TH)
6040 XD=X+8+12*COS(TH): YD=Y-7+12*SIN(TH)
6050 MOVE XD,YD: DRAW XE,YE,2
6060 '
6070  DG=INKEY(8): DD=INKEY(1)
6080  DH=INKEY(0): DB=INKEY(2): DTR=INKEY(47)
6090  IF DG*DD*DH*DB*DTR<>0 THEN T=T+0.2: GOSUB 6500:GOTO 6070
6100  TH=TH+((DG=0)-(DD=0)+7*((DH=0)-(DB=0)))*DT
6110  MOVE XD,YD: DRAW XE,YE,0
6120 IF DTR<>0 THEN 6020
6130 SOUND 1,478,5
6140 RETURN
6150 '
6500 '------------ SP AFFICHAGE TEMPS ------------------------
6510 '
6520 TAGOFF
6530 LOCATE #1, 4,2
6540 PRINT #1, INT(T);
6550 TAG
6560 RETURN
6570 '
7000 '------------ SP DEBUT COUP -----------------------------
7010 '
7020 LOCATE #1,13,2: PRINT #1,NB;
7030 '
7040 TAG: PEN 2
7050 XC=X1+10+INT(RND(1)*(X2-X1-20))
7060 YC=Y1+10+INT(RND(1)*(Y2-Y1-20))
7070 MOVE XC,YC,2: PRINT B$;
7080 '
7090 X=X1+10+LC+INT(RND(1)*(X2-X1-20-2*LC))
7100 Y=Y1+10+LC+INT(RND(1)*(Y2-Y1-20-2*LC))
7110 IF SQR((X-XC)^2+(Y-YC)^2)<=LC+10 THEN 7090
7120 MOVE X,Y,2: PRINT B$;
7130 '
7140 TC=0: NR=0
7150 GOSUB 6000
7160 CT=-DE*COS(TH): ST=-DE*SIN(TH)
7170 RETURN
7180 '
```

```
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 T=0: NT=0
8030 CLS #1
8040 LOCATE #1,3,1
8050 PRINT #1, "TEMPS    BOULE    COUPS-REUSSIS";
8060 RETURN
8070 '
8500 '------------ SP FIN DE PARTIE ------------------------
8510 '
8520 FOR K=1 TO 500: NEXT K
8530 SC=INT(NT*1000-5*T): IF SC<0 THEN SC=0
8540 LOCATE #1, 1,3
8550 PRINT #1, "SCORE";SC;
8560 PRINT #1, "M-SCORE";MS;
8570 IF SC>MS THEN MS=SC
8580 WHILE INKEY$<>"": WEND
8590 PRINT #1, "on rejoue";
8600 R$=INKEY$: IF R$="" THEN 8600
8610 RETURN
8620 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 NC=12: RX=2: MS=0
9030 BR=1: CT=7: CF=24: CB=6: CS=14
9040 X1T=3: X2T=38: Y1T=3: Y2T=20
9050 LC=80: DT=0.02: DE=16: TH=0
9060 '
9070 MODE 1
9080 SYMBOL AFTER 160
9090 SYMBOL 160,60,126,255,255,255,255,126,60
9100 B$=CHR$(160)
9110 '
9120 INK 0,CF: INK 1,CT: INK 2,CB: INK 3,CS
9130 WINDOW #1, 1,40,23,25
9140 PAPER #1,0: PEN #1,3: CLS #1
9150 '
9160 PAPER 1: CLS
9170 PAPER 0: BORDER BR
9180 FOR X=X1T TO X2T
9190  FOR Y=Y1T TO Y2T
9200   LOCATE X,Y: PRINT " ";
9210  NEXT Y
9220 NEXT X
9230 '
```

```
9240 X1=16*(X1T-1)+1: X2=16*X2T
9250 Y1=16*(25-Y2T): Y2=16*(26-Y1T)-1
9260 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

▷ **Pour modifier la couleur des boules et de la canne**

Remplacez, en 9030, la valeur 6 (dans CB = 6) par un nombre de votre choix entre 0 et 26.

▷ **Pour modifier la couleur du "tapis"**

Remplacez, en 9030 la valeur 24 (dans CF = 24) par un nombre de votre choix entre 0 et 26.

▷ **Pour modifier le nombre de coups par partie**

Remplacez, en 9020, la valeur 12 (dans NC = 12) par un nombre de votre choix.

▷ **Pour rendre le jeu plus facile ou plus difficile**

Vous pouvez agir sur le nombre obligatoire de rebonds. Pour cela, il vous suffit de remplacer, en 9020, la valeur 2 (dans RX = 2) par celle de votre choix. Ainsi :

- 0 conduira à un jeu extrêmement facile puisqu'aucun rebond ne sera nécessaire. Cela peut vous servir, au début, à vous entraîner.
- 1 conduira à un jeu assez facile ne demandant qu'un seul rebond.
- 2 correspond au programme actuel.
- 3 conduira à un jeu très difficile puisque trois rebonds seront nécessaires etc...

4. *Description du programme*

a) Techniques employées, décrites en annexe

— Hasard,

— Animation,

— Lecture rapide du clavier.

b) Le programme principal (100-270)

120 appelle le sous-programme d'initialisation du jeu.

130-260 correspondent au déroulement d'une partie ; elles sont répétées jusqu'à ce que vous précisiez que vous ne désirez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

140-240 répètent douze fois les instructions 150 à 230 qui constituent le jeu d'un coup :

150 appelle le sous-programme de ''début d'un coup'' (dessin des boules, mise en place de la canne et tir).

160-220 déplacent votre boule jusqu'à ce qu'elle touche la boule cible ou que le nombre de rebonds soit supérieur à deux :

160 détermine les nouvelles coordonnées de votre boule.

170 examine s'il y a contact entre les deux boules.

180 appelle, s'il y a lieu, le sous-programme ''rebond''.

190-210 déplacent votre boule.

230 appelle le sous-programme de fin d'un coup.

250 appelle le sous-programme de fin de partie.

c) Le sous-programme d'initialisation du jeu (9000-9260)

9020 fixe le nombre de coups d'une partie, le nombre de rebonds requis et initialise le meilleur score.

9030 détermine les couleurs.

9040 fixe les limites du ''tapis'' de jeu (en coordonnées texte).

9050 fixe la longueur de la canne, sa position initiale et les déplacements élémentaires.

9070-9100 définissent le caractère graphique représentant une boule.

9120-9140 définissent la fenêtre numéro 1 où s'affichent les scores et fixent les couleurs.

9160-9220 dessinent le billard.

9240-9250 déterminent les limites du billard (en coordonnées graphiques).

e) Le sous-programme de début d'un coup (7000-7170)

7020 affiche le numéro du coup.

7040-7070 déterminent, au hasard, la position de la boule cible et la dessinent.

7090-7140 tirent au hasard la position de votre boule (en tenant compte de la longueur de la canne) et la dessinent.

7140 initialise l'indicateur ''touché'' et le nombre de rebonds.

7150 appelle le sous-programme de mise en place de la canne.

7160 détermine les déplacements élémentaires de la boule.

f) Le sous-programme de mise en place de la canne (6000-6140)

6020-6120 sont répétées jusqu'à ce que vous pressiez la barre d'espace :

6020 actualise le temps et l'affiche (en appelant le sous-programme 6500).

6030-6050 dessinent la canne.

6070-6090 attendent qu'une touche du clavier soit pressée, en incrémentant le compteur de temps.

6100 détermine la nouvelle position (angulaire) de la canne.

6110 efface la canne de son ancienne position.

6120 examine si la touche pressée est la barre d'espace.

6130 émet un bruit de choc.

g) Le sous-programme de déplacement avec rebond (2000-2070)

2020-2050 déterminent la nouvelle position de la boule, incrémentent le compteur de rebonds et actualisent les déplacements élémentaires en fonction du bord sur lequel rebondit la boule.

2060 émet le son correspondant au rebond.

h) Le sous-programme de fin d'un coup (4000-4110)

4020 fait entendre un bruit de choc, s'il y a lieu.

4030 examine si le coup est réussi ou non.

4040-4070 en cas de coup réussi, font retentir un accord, incrémentent le score et l'affichent.

4080-4100 effacent les deux boules

i) Le sous-programme de fin de partie (8500-8610)

8520 attend un instant.

8530-8560 calculent le score et l'affichent en même temps que le meilleur score.

8570 actualise le meilleur score.

8580-8600 vous demandent si vous voulez rejouer.

5. Liste des variables

NC	Nombre de coups d'une partie.
RX	Nombre de rebonds requis.
MS	Meilleur score.
BR	Couleur du pourtour de l'écran.
CT	Couleur du bord du billard.
CF	Couleur du ''tapis'' et du fond où s'affichent les scores.
CB	Couleur des boules et de la canne.
CS	Couleur d'affichage des scores.
X1,X2,Y1,Y2	Coordonnées du ''tapis'' du billard (et coordonnées graphiques).
X1T, X2T, Y1T, Y2T	Coordonnées ''texte'' du tapis du billard.
LC	Longueur de la canne.
DT	Déplacement angulaire élémentaire de la canne.
DE	Déplacement élémentaire de la boule.
TH	Angle formé par la canne et l'axe horizontal.
T	Temps écoulé pendant la mise en place de la canne.
NT	Nombre de coups réussis.
XC, YC	Coordonnées de la boule cible.
X, Y	Coordonnées de votre boule.
TC	Indicateur ''boule cible touchée'' : 0 : pas de boule touchée 1 : boule cible touchée
NR	Compteur du nombre de rebonds, au cours d'un même coup.

CT	Déplacement horizontal élémentaire de votre boule.
ST	Déplacement vertical élémentaire de votre boule.
XE, YE	Coordonnées d'une extrêmité de la canne.
XD, YD	Coordonnées de l'autre extrêmité de la canne.
SC	Score.

Variables auxiliaires

R$ A   CL

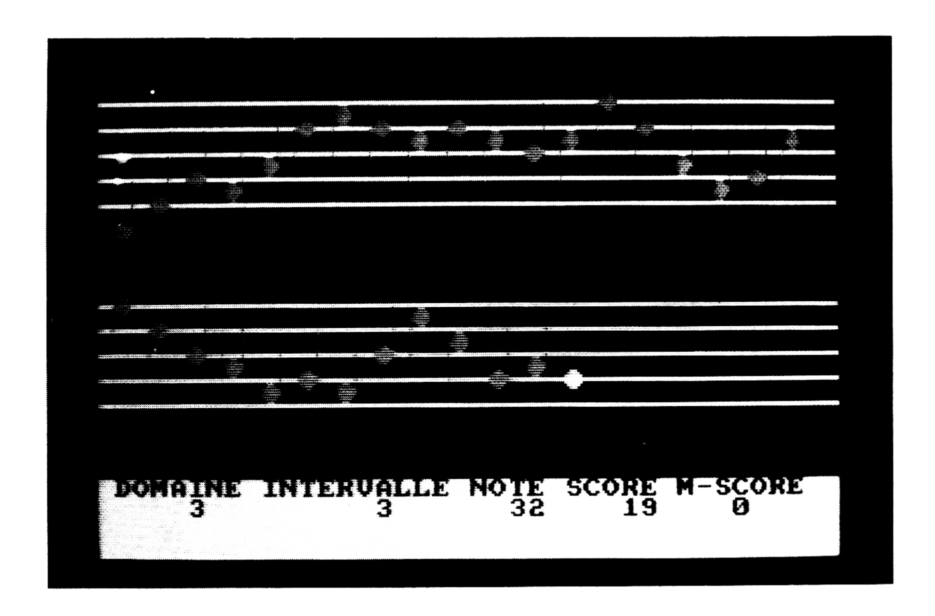
19
Dictée musicale

1. Le jeu

Voici un excellent moyen de révéler ou de développer vos talents de musicien. C'est en effet une ''dictée'' que vous propose le programme ; il va vous faire entendre des notes que vous devrez reconnaître en les plaçant sur une portée musicale. Sans aucun doute, la pratique répétée de ce jeu vous permettra d'affiner considérablement votre ''oreille''.

Au début du jeu apparaît une portée avec, en jaune, la note do (pour les ''initiés'', notez que nous sommes censés travailler en clé de sol, laquelle est la plus usuelle).

Le programme vous fait alors entendre le son correspondant, en transformant la note en rouge, pendant tout le temps où elle est jouée. Une deuxième note vous est alors jouée. Il vous faut la reconnaître en plaçant correctement une note sur la portée. Pour cela, vous déplacez la dernière note (blanche) dessinée par le pro-



gramme (à un endroit toujours inexact) à l'aide des touches "↑" et "↓". Quand vous pensez l'avoir bien située, vous appuyez sur la barre d'espace. Si votre réponse est correcte, vous marquez un point. Dans le cas contraire, votre note est effacée tandis qu'un son approprié retentit et la note correcte est dessinée.

Le programme va alors vous faire deviner une troisième note, en vous faisant entendre à nouveau les deux notes qui figurent déjà sur la portée. A chaque fois, la note jouée se signale en "rougissant".

Le jeu se poursuit ainsi, de note en note, de devinette en devinette. Toutefois, pour que tout cela ne devienne pas trop lent, à partir de la sixième note, le programme se contente de vous faire entendre les cinq dernières notes seulement. Cela devrait vous fournir une préparation sonore suffisante pour trouver la dernière note entendue. Si ce n'était pas le cas, vous verrez au paragraphe 3, comment modifier cela.

Le programme vous propose 27 niveaux de jeu, grâce à deux paramètres que vous pouvez choisir :

a) Le domaine dans lequel vont évoluer les notes. Trois options possibles :

1 — de "do" à "sol" : cela fait donc 5 notes en tout : do, ré, mi, fa et sol.

2 — de "do" à "do" : cela correspond donc aux 8 notes d'un octave.

3 — de "do" à "la" : il faut comprendre du do d'un octave au la de l'octave supérieur. Cela correspond donc à 13 notes.

b) L'intervalle séparant deux notes successives. Il peut varier de un à neuf.

A titre d'exemple, si vous choisissez la valeur 1, la note émise après un mi ne pourra être que ré ou fa.

Par contre, si vous choisissez la valeur 3, la note émise après un mi pourra être : ré, do, fa ou sol.

Le programme affiche en permanence le numéro du domaine choisi, l'intervalle choisi, le numéro de la note à deviner (nombre de notes déjà jouées), votre score (nombre de notes reconnues) et le meilleur score du niveau.

2. Le programme

```
100 '******************* DICTEE MUSICALE ***********************
110 '
120 GOSUB 9000
130  GOSUB 8000
140  NN=1: NT=1: C=2: GOSUB 4000: GOSUB 5000
150  NJ(1)=1
160 '
170  FOR N=2 TO NX
180   LOCATE #1, 22,2: PRINT #1,N;
190   ND=N-5: IF ND<1 THEN ND=1
200   FOR NN=ND TO N-1
210    NT=NJ(NN)
220    C=3: GOSUB 4000: GOSUB 3000
230    C=2: GOSUB 4000
240   NEXT NN
250 '
260   CX=1:IF RND(1)>0.5 THEN CX=-1
270   NT=NJ(N-1)+CX*INT(RND(1)*IM+1)
280   IF NT<1 OR NT>NM THEN 260
290   NJ(NN)=NT
300   GOSUB 3000
310 '
320   GOSUB 7000
330   IF NT=NJ(NN) THEN SC=SC+1 ELSE SOUND 1,3000,25
340   C=2: NT=NJ(N): GOSUB 4000: GOSUB 5000
350   LOCATE #1,28,2: PRINT #1,SC;
360  NEXT N
370 '
```

```
380  IF SC>MS(DM,IM) THEN MS(DM,IM)=SC
390  WHILE INKEY$<>"": WEND
400  LOCATE #1,3,3: PRINT #1,"rejouez vous?";
410  R$=INKEY$: IF R$="" THEN 410
420 IF R$<>"N" THEN 130
430 END
440 '
2000 '------------ SP CALCUL COORDONNEES NOTE-----------------
2010 '
2020 IF NN<=NX/2 THEN YB=Y1: X=X1+(NN-1)*DX
2030 IF NN>NX/2 THEN YB=Y2: X=X1+(NN-NX/2-1)*DX
2040 Y=YB+(NT-1)*(DT+ET)/2+8
2050 RETURN
2060 '
3000 '------------ SP SON NOTE HAUTEUR NT --------------------
3010 '
3020 SOUND 1,NT(NT),30
3030 FOR K=1 TO 300: NEXT K
3040 RETURN
3050 '
4000 '------------ SP DESSIN ROND COULEUR C ------------------
4010 '
4020 GOSUB 2000
4030 MOVE X,Y,C
4040 PRINT NT$;
4050 RETURN
4060 '
5000 '------------ SP AFFICHAGE QUEUE -----------------------
5010 '
5020 IF NT<=7 THEN MOVE X+15,Y-8,C: DRAW X+15,Y+32
5030 IF NT>7 THEN MOVE X-2,Y-8,C: DRAW X-2,Y-40
5040 RETURN
5050 '
6000 '------------ SP EFFACEMENT NOTE EN X,Y ----------------
6010 '
6020 MOVE X,Y,0: PRINT NT$;
6030 IF (NT-3)*(NT-5)*(NT-7)*(NT-9)*(NT-11)<>0 THEN RETURN
6040 FOR YT=Y-6-ET/2 TO Y-8+ET/2
6050  MOVE X,YT,1: DRAW X+16,YT
6060 NEXT YT
6070 RETURN
6080 '
```

```
7000 '------------ SP LECTURE NOTE PROPOSEE-------------------
7010 '
7020 WHILE INKEY$<>"": WEND
7030 NT=NJ(N-1)
7040  GOSUB 2000
7050  C=1: GOSUB 4000
7060  R$=INKEY$: IF R$="" THEN 7060
7070  GOSUB 6000
7080  R=ASC(R$)
7090  NT=NT+(R=240)*(NT<NM)-(R=241)*(NT>1)
7100 IF R<>32 THEN 7040
7110 RETURN
7120 '
8000 '------------ SP INITIALISATION PARTIE -----------------
8010 '
8020 TAGOFF: CLS
8030 LOCATE 5,5
8040 PRINT "D I C T E E   M U S I C A L E";
8050 LOCATE 3,10: PRINT "3 domaines possibles";
8060 LOCATE 5,11: PRINT "1 : DO a SOL";
8070 LOCATE 5,12: PRINT "2 : DO a DO";
8080 LOCATE 5,13: PRINT "3 : DO a LA";
8090 LOCATE 3,15: PRINT "domaine choisi?";
8100 R$=INKEY$: IF R$="" THEN 8100
8110 DM=VAL(R$): IF DM<1 OR DM>9 THEN 8100
8120 PRINT DM;
8130 '
8140 LOCATE 3,20: PRINT "intervalle maximum entre";
8150 LOCATE 3,21: PRINT "2 notes (1 a 9)?";
8160 R$=INKEY$: IF R$="" THEN 8160
8170 IM=VAL(R$): IF IM<1 OR IM>9 THEN 8160
8180 PRINT IM;
8190 FOR K=1 TO 500: NEXT K
8200 '
8210 CLS: TAG
8220 FOR YB=Y2 TO Y1 STEP Y1-Y2
8230  FOR I=1 TO 5
8240   Y=YB+I*(DT+ET)
8250   FOR J=1 TO ET
8260    MOVE XD,Y+J-1,1
8270    DRAW XF,Y+J-1
8280   NEXT J
8290  NEXT I
8300 NEXT YB
8310 '
```

```
8320 SC=0
8330 NM=5: IF DM=2 THEN NM=8
8340 IF DM=3 THEN NM=13
8350 CLS #1
8360 LOCATE #1, 2,1
8370 PRINT #1, "DOMAINE INTERVALLE NOTE SCORE M-SCORE";
8380 LOCATE #1, 5,2: PRINT #1, DM;
8390 LOCATE #1, 15,2: PRINT #1, IM;
8395 LOCATE #1, 34,2: PRINT #1,MS(DM,IM);
8400 RETURN
8410 '
9000 '------------ SP INITIALISATION DU JEU -----------------
9010 '
9020 CE=0: CP=26: BR=0: CND=24: CNP=6
9030 X=1: XF=640: X1=17
9040 Y1=260: Y2=100
9050 DT=16: ET=4: DX=32
9060 NX=2*INT((XF-X1)/DX)
9070 '
9080 DIM NJ(NX), NT(13), MS(3,9)
9090 DATA 478,426,379,358,319,284,253
9100 DATA 239,213,190,179,159,142
9110 FOR I=1 TO 13: READ NT(I): NEXT I
9120 '
9130 SYMBOL AFTER 160
9140 SYMBOL 160, 60,126,255,255,255,255,126,60
9150 NT$=CHR$(160)
9160 '
9170 INK 0,CE: INK 1,CP: INK 2,CND: INK 3,CNP
9180 WINDOW #1, 1,40,22,25
9190 PAPER #1,1: PEN #1,3
9200 PAPER 0: PEN 1: BORDER BR
9210 RETURN
```

3. *Si vous souhaitez personnaliser ce programme*

▷ **Si vous trouvez que le jeu est trop lent ou trop rapide**

Vous pouvez modifier la durée de chaque note en remplaçant, en 3030, la valeur 300 par un nombre de votre choix. Ainsi, avec :

```
3030 FOR K = 1TO 150 : NEXT K
```

les notes seront deux fois plus brèves.

▷ **Pour modifier le nombre de notes jouées avant la nouvelle note à deviner**

Remplacez, en 190, le nombre 5 (dans ND = N − 5) par une valeur de votre choix. Elle représentera le nombre de notes jouées avant celle à deviner. Par exemple avec :

```
190 ND=N-1 : IF ND<1 THEN ND=1
```

le programme ne jouera qu'une seule note avant celle à deviner.

Par contre, avec :

```
190 ND=N-10 : IF ND<1 THEN ND=1
```

le programme jouera, à chaque fois, dix notes (du moins, à partir du moment où il y en aura déjà dix de représentées).

Enfin, si vous souhaitez que *toutes* les notes soient systématiquement rejouées avant la note à deviner, remplacez 190 par :

```
190 ND=1
```

4. Description du programme

a) Techniques employées, décrites en annexe

— Hasard,

— Animation.

b) Le programme principal (100-430)

120 appelle le sous-programme d'initialisation du jeu.

130-420 correspondent au déroulement d'une partie ; elles sont répétées jusqu'à ce que vous précisiez que vous ne souhaitez plus rejouer :

130 appelle le sous-programme d'initialisation d'une partie.

140-150 dessinent et jouent la première note et placent son numéro (1) dans le tableau des notes déjà jouées.

170-360 répètent les instructions qui proposent, à chaque fois une nouvelle note à découvrir :

180 affiche le numéro de la note à découvrir.

190-240 font entendre les cinq dernières notes (ou toutes les notes lorsqu'il y en a moins de cinq) en les faisant ''rougir''.

260-290 choisissent au hasard une nouvelle note (en tenant compte des limitations imposées par votre choix de domaine et d'intervalle) et rangent sa valeur dans le tableau NJ.

300 fait retentir la note ainsi déterminée.

320 appelle le sous-programme qui ''lit'' votre proposition.

330 analyse votre réponse ; si elle est correcte, le score est actualisé, sinon un ''zap'' est émis.

340 dessine la note correcte (quelle que soit votre réponse).

350 affiche le score.

380 actualise le meilleur score.

390-400 vous demandent si vous souhaitez rejouer.

410 lit votre réponse.

c) Le sous-programme d'initialisation du jeu (9000-9210)

9020 détermine les couleurs.

9030 fixe les limites horizontales des portées et l'abscisse correspondant à la première note.

9040 fixe les ordonnées de la ''base'' des deux portées.

9050 fixe la distance entre deux traits successifs d'une portée, l'épaisseur d'un trait de portée et la distance horizontale entre deux notes successives.

9060 calcule le nombre total de notes qui seront proposées.

9080 réserve les tableaux.

9090-9110 fixent les notes employées.

9130-9150 fabrique le caractère représentant une note.

9170-9200 définissent la fenêtre numéro 1 où s'afficheront les scores et fixent les couleurs.

d) Le sous-programme d'initialisation d'une partie (8000-8400)

8020-8040 affichent le titre.

8050-8120 vous font choisir votre ''domaine''.

8130-8190 vous font choisir l'intervalle maximum entre deux notes et attendent un instant.

8210-8300 dessinent les deux portées.

8320 initialise le score.

8330-8340 déterminent la note la plus haute, compte tenu du domaine choisi.

8350-8400 affichent les ''titres'' des différents scores, le domaine et l'intervalle maximum.

e) Le sous-programme d'affichage d'un ''rond'' de couleur C (4000-4050)

Il affiche la note (sans queue) dont la position sur la portée est définie par NN et NT.

4020 appelle le sous-programme déterminant les coordonnées de la note.

4030-4040 dessinent la note (sans queue) dans la couleur C.

f) Le sous-programme de lecture de la note proposée (7000-7110)

7030 définit la hauteur de la note (blanche) à afficher comme étant celle de la dernière note jouée.

7040 appelle le sous-programme de détermination des coordonnées de cette note.

7050 dessine cette note (en blanc).

7060 attend que vous frappiez une touche.

7070 efface la note blanche.

7080-7090 déterminent la ''hauteur'' de la note blanche, compte tenu du déplacement demandé.

7100 examine si la barre d'espace a été pressée.

5. Liste des variables

CE	Couleur de fond de la zone où apparaissent les portées.
CP	Couleur des traits des portées et du fond de la fenêtre d'affichage des scores.
BR	Couleur du tour de l'écran.
CND	Couleur représentant une note définitive (ici jaune).
CNJ	Couleur prise par une note, tandis qu'elle est jouée par le programme (ici rouge) et couleur des scores.

XD,XF	Abscisses (graphiques) de début et de fin des deux portées.
X1	Abscisse de la première note.
Y1	Ordonnée ''de base'' de la première portée.
Y2	Ordonnée ''de base'' de la seconde portée.
DT	Intervalle séparant deux lignes consécutives d'une portée.
ET	Épaisseur des lignes des portées.
DX	Écart entre les abscisses de deux notes successives.
NX	Nombre total de notes jouées au cours d'une partie.
NJ(NX)	Tableau contenant les hauteurs des notes déjà jouées.
NT(13)	Tableau contenant les treize notes possibles.
MS(3,9)	Tableau des meilleurs scores.
DM	Domaine choisi (1 à 3).
IM	Intervalle maximum choisi (1 à 9).
YB	Ordonnée ''de base'' de l'une des deux portées.
SC	Score.
NM	Hauteur de la plus haute note possible, compte tenu du domaine choisi.
NN	Numéro de note ''courante''.
NT	Hauteur de note ''courante''.
C	Couleur de note ''courante''.
N	Numéro de la dernière note proposée.
ND	Numéro de la note à partir de laquelle s'effectue la répétition qui précède l'émission de la dernière note.
X, Y	Coordonnées d'une note courante.

Variables auxiliaires

R$ A I

Y CX R

YT

Annexe 1
Hasard

Bien qu'aucun des jeux de ce livre ne soit ce que l'on a coutume d'appeler des ''jeux de hasard'', il se trouve que le hasard intervient dans la plupart d'entre eux. C'est par exemple le cas lorsqu'il faut choisir ''au hasard'' la position d'un objet sur l'écran, une lettre de l'alphabet, etc...

Comment choisir une valeur au hasard

La fonction RND fournit une valeur quelconque comprise entre 0 (inclu) et 1 (exclu). Généralement, nous avons besoin d'une valeur *entière* comprise entre deux valeurs données que nous noterons n et p.

■ Voyons déjà comment procéder quand n vaut 1, autrement dit lorsque l'on désire obtenir une valeur entière comprise entre 1 et n (1 et n compris).

Expressions	*Valeurs correspondantes*
RND (1)	entre 0 (compris) et 1 (exclu)
n*RND (1)	entre 0 (compris) et n (exclu)
n*RND (1) + 1	entre 1 (compris) et n+1 (exclu)
INT(n*RND (1) + 1)	entière entre 1 (compris) et n (compris).

Quelques exemples :

- pour tirer un dé (nombre entre 1 et 6) dans X :

  X=INT(6*RND(1)+1)

- pour tirer une couleur d'encre, entre 0 et 3, dans C :

  C=INT(4*RND(1))

■ Voyons maintenant le cas plus général évoqué précédemment : obtenir un nombre entier compris entre p (compris) et n (compris) :

Expressions	*Valeurs correspondantes*
RND (1)	entre 0 (compris) et 1 (exclu)
(n-p+1)*RND (1)	entre 0 (compris) et n-p+1 (exclu)
p+(n-p+1)*RND (1)	entre p (compris) et n+1 (exclu)
INT(p+(n-p+1)*RND (1))	entière entre 1 (compris) et n (compris).

La dernière expression peut également s'écrire :

  p+INT((n-p+1)*RND (1))

Quelques exemples :

- pour tirer dans X une coordonnée d'écran comprise entre 2 limites X1 et X2 :

  X=X1 + INT(RND (1)*(X2-X1+1))

- pour tirer dans C le code ASCII d'une lettre comprise entre A (code 65) et Z (code 90) :

  C=65+INT(RND (1)*26)

Annexe 2
Animation

1. Le principe

Pour donner l'impression qu'un objet se déplace sur l'écran, une méthode consiste à le faire apparaître un instant en un emplacement donné, l'effacer puis le faire apparaître en un emplacement voisin, et ainsi de suite. Si le rythme ainsi créé est suffisamment rapide, l'objet donnera l'impression de se déplacer de manière continue.

Voici à titre d'exemple, un petit programme déplaçant le caractère "*" de gauche à droite de l'écran :

```
100 Y=12
110 FOR X=1 TO 40
120       LOCATE X,Y : PRINT "*" ;
130       FOR K=1 TO 100 : NEXT K
140       LOCATE X,Y : PRINT "  "
150 NEXT X
```

Notez l'instruction 130 qui permet de réaliser une courte attente. Dans la pratique, il y aura suffisamment de choses à faire, entre l'apparition et la disparition du mobile, pour qu'jl ne soit plus nécessaire de ralentir le rythme ainsi obtenu.

2. *Les limitations du déplacement*

Lorsqu'un mobile se déplace sur l'écran, il est nécessaire d'imposer des restrictions à son déplacement :

- pour éviter des collisions ou pour vérifier qu'une cible est atteinte. Cela nécessite d'examiner le contenu du prochain emplacement prévu afin de prendre la décision adéquate.
- pour éviter qu'il ne sorte de certaines limites qu'on lui impose. Il peut s'agir de limites d'un domaine de jeu ou tout simplement de celles de l'écran. N'oubliez pas, par exemple, que l'instruction :

```
LOCATE X,Y
```

ne peut s'exécuter qu'avec des valeurs de X comprises entre 1 et 40 et des valeurs de Y comprises entre 1 et 25.

La plupart du temps, pour résoudre de tels problèmes, vous serez amenés à considérer deux jeux de coordonnées pour chaque mobile. Par exemple, dans nos programmes, nous avons souvent utilisé :

- X,Y pour les coordonnées du mobile à un instant donné.
- XN,YN pour les coordonnées du prochain emplacement prévu pour le mobile. Leur calcul se fait à partir des valeurs de X,Y, d'éventuelles limitations et, s'il y a lieu, en fonction de commandes de déplacement transmises par le clavier (touches fléchées par exemple).

A chaque pas de déplacement, on calcule XN et YN, puis on examine si le mobile peut s'y déplacer. Dans ce cas, on efface le contenu de l'emplacement de coordonnées X et Y ; puis on dessine le mobile en XN,YN en actualisant les valeurs de X et Y par les instructions.

```
X=XN
Y=YN
```

Annexe 3
Lecture du clavier

En Basic AMSTRAD, il existe une instruction et deux fonctions qui permettent ''d'accéder'' au clavier : INPUT, INKEY$ et INKEY. Comme nous allons le voir, l'instruction INPUT n'est pas du tout adaptée à la conduite de mobiles par l'intermédiaire du clavier (déplacement d'une base de tir, d'une raquette, d'une soucoupe, etc...). La fonction INKEY$, par contre, sera utilisable dans certaines circonstances particulières. Dans les autres cas, il sera nécessaire de recourir à la fonction INKEY comme nous le montrerons dans l'annexe 4 : lecture rapide du clavier.

L'instruction INPUT

***INPUT** lit une chaîne de caractères en l'affichant à l'écran en même temps que vous la composez. Son action se termine lorsque vous frappez la touche ''entrée''. Cette instruction qui ''bloque'' le programme tant que vous n'avez pas fini de frapper votre message, est inutilisable pour commander des déplacements.

La fonction INKEY$

Elle se contente d'examiner si une touche du clavier est enfoncée. Si c'est le cas, elle restitue le caractère correspondant ; dans le cas contraire, elle fournit une "chaîne vide". En apparence, cette fonction devrait résoudre les problèmes de contrôle de mobile puisqu'elle ne bloque plus le programme. Malheureusement, une imperfection de taille apparaît. Pour vous en convaincre, essayez ce petit programme :

```
100 CLS : LOCATE 1,10
110 FOR I=1 TO 30
120      A$=INKEY$ : IF A$=""THEN 120
130      PRINT A$ ;
140 NEXT I
```

Vous constatez que, tant que vous n'appuyez sur aucune touche, il ne se passe rien : l'instruction 120 "boucle sur elle-même". Si vous pressez, par exemple, la touche F, le programme vous écrit immédiatement F. Par contre, si vous laissez cette touche enfoncée en permanence, vous découvririez qu'il s'écoule un temps assez long avant l'apparition du second F ; puis les lettres suivantes arrivent ensuite à un rythme régulier et plus rapide.

En fait, vous observez le même phénomène que lorsque vous tapez plusieurs fois la même lettre, grâce à la "répétition automatique".

Nous n'entrerons pas dans le détail du fonctionnement de INKEY$; nous retiendrons seulement qu'il est tributaire du rythme de la répétition automatique. Son emploi rend alors désagréable la commande de déplacement d'un mobile par les quatre touches fléchées. Il existe toujours quelques circonstances où ce problème de rythme n'apparaît pas ; c'est notamment le cas lorsque les touches servent, non pas à commander chaque déplacement du mouvement, mais simplement à des changements de direction. C'est également le cas lorsque l'on peut se permettre un déplacement lent ou irrégulier (exemple : drôles de mines).

La commande des déplacements par INKEY$

On peut utiliser n'importe quelle touche du clavier pour commander le mouvement. Si l'on convient, par exemple, que la touche P correspond à la direction "vers le haut" de l'écran, on pourrait "tester" l'enfoncement de cette touche par :

```
A$=INKEY$ : IF A$="P" THEN...
```

Toutefois, il paraît plus judicieux d'employer la touche fléchée "↑". Dans ces conditions, le test ne peut plus se faire par comparaison directe de A$ avec un caractère donné, car la touche "↑" ne correspond pas à un carractère représentable (il en va de même de "entrée", "raz", →, ↓, ←, etc...). Pour manipuler ces caractères, il est nécessaire d'en connaître le "code ASCII".

Voici les codes ASCII correspondant aux quatre touches fléchées :

Touches	*Code*
←	242
→	243
↑	240
↓	241

Pour savoir si la touche "↑" a été pressée, il suffira :

— soit de comparer A$ avec CHR$(240) :

```
IF A$=CHR$(240) THEN...
```

— soit de comparer avec 240 le code ASCII du caractère contenu dans A$:

```
IF ASC(A$)=240 THEN...
```

Exemple : déplacement lent

Soit un mobile ayant, à un instant donné les coordonnées X et Y. On souhaite le déplacer à l'aide des touches fléchées. Voici les instructions permettant de calculer les nouvelles coordonnées XN, YN, en fonction de la touche pressée (notez qu'ici on attend qu'une touche soit pressée).

```
180 D$=INKEY$ : IF D$="" THEN 180
190 D=ASC(D$)
200 XN=X + (D=242) - (D=243)
210 YN=Y + (D=240) - (D=241)
```

N'oubliez pas qu'une expression "logique" telle que D = 240 prend la valeur – 1 lorsqu'elle est vraie et la valeur 0 lorsqu'elle est fausse.

Dans cet exemple, nous n'avons pas prévu de limitations du déplacement, ce qui sera nécessaire en pratique. Ainsi, si l'abscisse du mobile doit rester comprise entre deux limites X1 et X2, l'instruction 200 peut être remplacée par :

```
200 XN=X -(D=242) * (X > X1) + (D=243) * (X < X2)
```

REMARQUE : la méthode que nous venons de décrire est utilisée dans les jeux suivants : drôles de mines, embarquement immédiat, flip-flop, dictée musicale.

Exemple de commande de changement de direction

Soit un mobile ayant, à un instant donné les coordonnées X et Y et une direction de déplacement DL codée ainsi :

- DL = 242 : déplacement vers la gauche,
- DL = 243 : déplacement vers la droite,
- DL = 241 : déplacement vers le bas,
- DL = 240 : déplacement vers le haut.

Nous souhaitons déterminer la nouvelle direction, puis la nouvelle position du mobile, en tenant compte d'une touche éventuellement pressée. Si aucune touche n'est enfoncée, la direction reste inchangée, et le calcul des nouvelles coordonnées se fait de façon usuelle, par exemple :

```
XN = X + (DL=242) - (DL = 243)
```

Si une touche a été enfoncée, il faut en tenir compte pour modifier DL, à condition que cette touche soit bien l'une des quatre flèches ; cette vérification se fait en s'assurant que le code ASCII correspondant est compris entre 240 et 243.

Voici une façon de résoudre ce problème, en créant une valeur intermédiaire DN, contenant le code de la touche pressée. Par convention, nous y plaçons la valeur zéro quand aucune touche n'est frappée :

```
R$=INKEY$ : IF R$="" THEN DN=0 ELSE DN=ASC(R$)
IF DN>=240 AND DN<=243 THEN DL=DN
XN=X + (DL=242) - (DL=243)
YN=Y + (DL=240) - (DL=241)
```

Là encore, il est généralement nécessaire de limiter le déplacement.

REMARQUE : cette méthode est employée dans les jeux suivants : reconstitution, karting.

La mémoire du clavier

Nous avons dit que INKEY$ fournissait le caractère correspondant à la touche pressée au moment où l'on "exécutait" cette fonction. En fait, ce n'est pas toujours exact dans la mesure où votre Amstrad dispose d'une "mémoire de clavier" dans laquelle il mémorise la succession des touches que vous frappez (jusqu'à concurrence de 20). La fonction INKEY$ fournit, non pas le dernier caractère frappé, mais le plus ancien mémorisé (il est alors naturellement "éliminé" de la mémoire clavier). Certes, il s'agit souvent de la même chose mais voici un petit programme qui va vous prouver qu'il n'en est pas toujours ainsi.

```
100 FOR I=1 TO 5000 : NEXT I
110 FOR I=1 TO 10
120       PRINT INKEY$
130 NEXT I
```

La ligne 100 réalise une ''boucle d'attente'' de quelques secondes. Exécutez ce programme en tapant sur quelques touches pendant l'attente. Vous constatez que, tout d'abord, rien ne s'affiche à l'écran. Puis, au bout de quelques secondes, tous les caractères frappés (ou du moins les 10 premiers) apparaissent, à raison d'un par ligne.

Ce mécanisme de ''mémoire de clavier'' peut présenter des avantages et des inconvénients.

Un des avantages tient à ce que vous pouvez frapper certaines choses à l'avance. Encore faut-il que le jeu s'y prête !

Le principal inconvénient réside dans le risque de ''lire'' quelque chose qui a été tapé plus tôt. Ce sera le cas, par exemple, lorsque le programme vous demandera si vous souhaitez rejouer et qu'il prendra comme réponse votre dernière commande de déplacement ! il est toutefois facile de résoudre ce problème en commençant par ''vider'' la mémoire clavier par des instructions telles que :

```
100 R$=INKEY$ : IF R$ <> "" THEN 100
```

ou

```
100 WHILE INKEY$ <> "" : WEND
```

Annexe 4
Lecture rapide du clavier

Dans l'annexe ''lecture du clavier'', nous avons vu que la fonction INKEY$ ne permettait de réaliser que certains types de déplacements de mobiles. Notamment, elle ne convient pas aux déplacements rapides et réguliers.

La fonction INKEY

Malgré ses défauts, la fonction INKEY$ avait un énorme avantage : celui de nous fournir le caractère frappé, *quel que soit ce caractère*.

La fonction INKEY ne présente aucun des inconvénients de INKEY$ mais elle ne permet de recueillir des informations que pour une touche donnée. Plus précisément, si l'on souhaite examiner 10 touches différentes, il faudra employer 10 fois cette fonction.

Chaque touche porte un numéro ; citons :

Touche	*Numéro*
barre d'espace	47
←	8
→	1
↑	0
↓	2

(Attention, ce n'est pas le caractère ASCII d'un des caractères correspondant à cette touche).

L'expression INKEY (47) représente ''l'état'' de la touche de numéro 47 ; INKEY (65) sera l'état de la touche numéro 65. Cet état est ''codé'' de cette façon :

- 1 signifie que la touche n'est pas enfoncée
  0 signifie que la touche est enfoncée

Il existe d'autres états correspondants aux cas où SHIFT ou CTL sont enfoncés en même temps que la touche concernée. Nous n'en aurons pas besoin ici.

Pour savoir, par exemple, si la touche ← est enfoncée, on procèdera ainsi :

```
IF INKEY (8)=0 THEN...
```

2. Exemple de commande de déplacement

Soit un mobile ayant, à un instant donné, les coordonnées X et Y. On souhaite le déplacer à l'aide des touches fléchées. Voici les instructions permettant d'en calculer les nouvelles coordonnées XN,YN en fonction de la touche pressée :

```
200 DG=INKEY(8) : DD=INKEY(1)
210 DH=INKEY(0) : DB=INKEY(2)
220 XN=X + (DG=0) - (DD=0)
230 YN=Y + (DH=0) - (DB =0)
```

Là encore, il sera souvent nécessaire, en pratique, de limiter le déplacement. Ainsi, par exemple, pour imposer que l'abscisse du mobile reste comprise entre X1 et X2, la ligne 220 pourra être remplacée par :

```
220 XN=X - (DG=0) * (X > X1) + (DD=0) * (X < X2)
```

REMARQUE : cette méthode est employée, avec éventuellement quelques variantes, dans les jeux suivants : goal, mouches, ballons, briques, envahisseurs, billard.

Annexe 5
Examen du contenu de l'écran

Dans de nombreux jeux, il est nécessaire de connaître, à un instant donné, ce qui se trouve en un certain emplacement de l'écran. Par exemple, vous pouvez vouloir vérifier qu'un bolide peut se déplacer d'une position, sans heurter un mur. Ou encore, vous avez besoin de savoir si le prochain emplacement d'un missile correspond à une case vide ou, au contraire, à l'une des cibles à atteindre.

En fait, le Basic de votre AMSTRAD ne vous permet pas de connaître le caractère affiché à un emplacement texte de coordonnées données. Par contre, il vous permet, dans une certaine mesure, d'en connaître la couleur, grâce à la fonction TEST. Néanmoins, cette dernière utilise les coordonnées graphiques, ce qui impose une petite ''gymnastique'' lorsque le reste du programme travaille en ''coordonnées texte''.

Correspondance entre coordonnées texte et coordonnées graphiques

Notons tout d'abord que les coordonnées texte (du moins dans le sens horizon-

tal) dépendent du mode choisi. Il n'en va pas de même pour les coordonnées graphiques. Considérons le cas le plus fréquent du mode numéro 1.

Chaque caractère y est représenté par un ''carré'' de 16 × 16 points. Le schéma de la page suivante représente le coin inférieur gauche de l'écran, en précisant les deux types de coordonnées.

On peut montrer que tous les points graphiques d'un emplacement texte de coordonnées X,Y ont des coordonnées comprises :

— entre 16 * (X − 1) + 1 et 16 * (X − 1) + 16 pour les abscisses

— entre 16 * (25 − Y) + 1 et 16 * (25 − Y) + 16 pour les ordonnées

La fonction TEST

Elle fournit le *numéro d'encre* correspondant à un point de coordonnées graphiques données. En mode 1, ce numéro est donc compris entre 0 et 3.

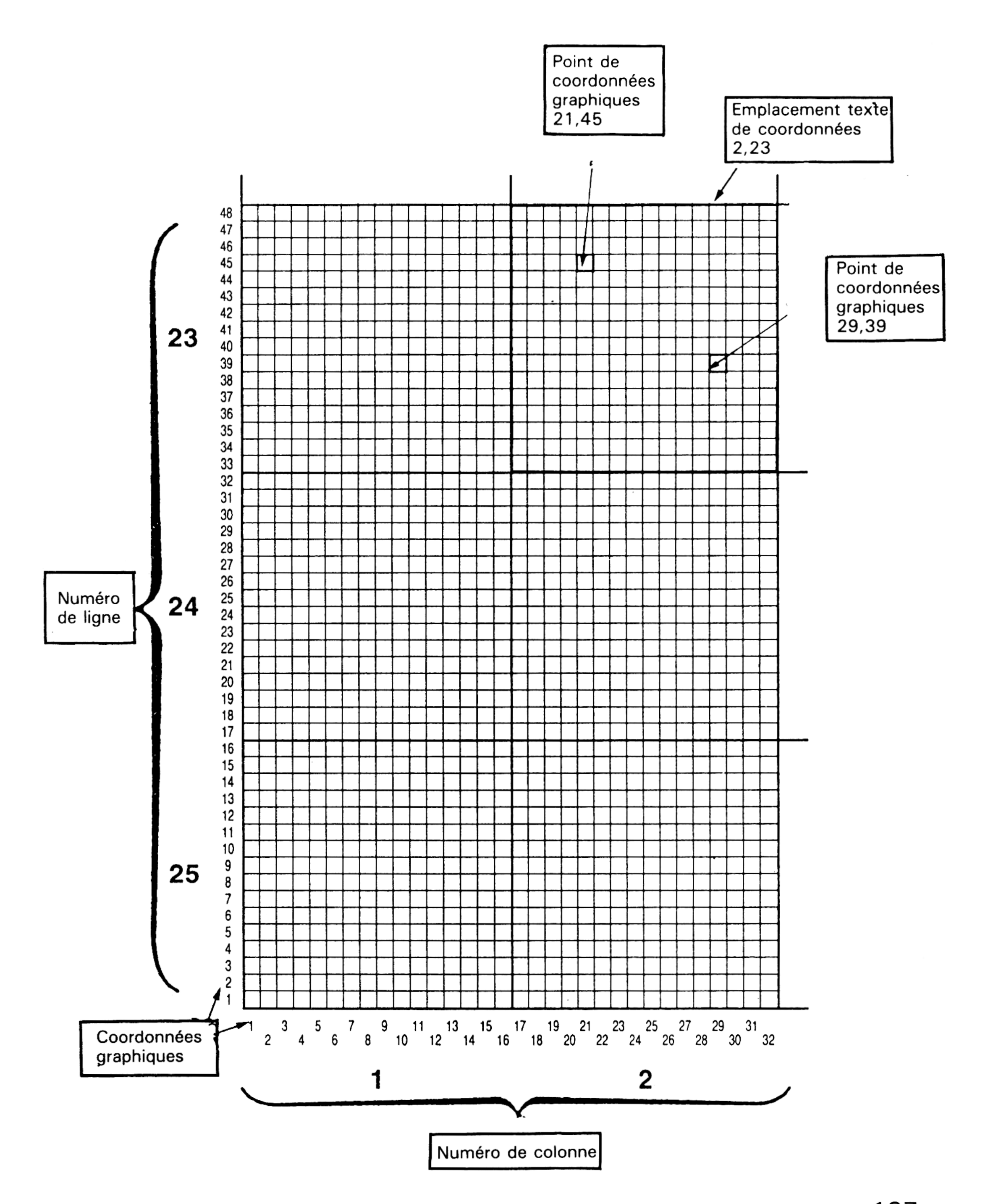
Or, la plupart du temps, nous avons besoin, non pas de connaître la couleur d'un point graphique donné, mais plutôt de répondre à des questions telles que :

- L'emplacement texte X,Y est-il disponible ?
- Quel est le caractère affiché à l'emplacement texte X,Y ?

A priori, on pourrait penser que de telles questions nécessitent d'examiner tous les points graphiques correspondants à un emplacement texte. Cela conduirait à 16 × 16 = 256 tests ! (en toute rigueur 8 × 8 suffiraient, compte tenu de la ''résolution'' du mode 1).

En pratique, on s'arrange pour n'avoir à considérer qu'un seul point, bien choisi. Ainsi, en ce qui concerne la première question, il n'y aura généralement qu'un éventail restreint de caractères susceptibles d'être présents en un emplacement donné. On n'examinera alors qu'un seul point graphique dont on est sûr qu'il appartient à tous les dessins des caractères que l'on cherche à détecter.

Quant à la seconde question, on ne pourra y répondre de façon simple que lorsque le nombre de caractères possibles est suffisamment petit pour que chacun possède sa propre couleur. Là encore, on choisira un point graphique appartenant aux différents caractères. Sa couleur nous renseignera alors sur la nature du caractère examiné.


Point de coordonnées graphiques 21,45
Emplacement texte de coordonnées 2,23
Point de coordonnées graphiques 29,39
Numéro de ligne
23
24
25
Coordonnées graphiques
1
2
Numéro de colonne

Composition Compo 2000 - Saint - Lô
Imprimerie de la Manutention à Mayenne
Dépôt légal : Avril 1986
N° d'éditeur : 4467

Vous aimez les jeux sur mesure ! Vous souhaitez aller au-delà de la simple consommation passive de produits figés ! Alors ce livre vous est destiné, et ceci quel que soit votre degré de pratique de la programmation en Basic.

Comme il se doit, il vous offfre tout d'abord un éventail de jeux variés (classiques ou originaux) exploitant pleinement les formidables capacités graphiques et sonores de votre Amstrad. Suivant votre humeur, vous pourrez ainsi affiner vos réflexes, exercer votre mémoire, améliorer vos capacités de raisonnement ou encore vous former une « oreille » de musicien.

Mais ensuite, et surtout, il vous offre l'opportunité d'adapter chaque jeu en fonction de vos goûts et vos désirs. Il vous suffit pour cela de choisir parmi les nombreuses propositions de personnalisation. Elles sont expliquées très clairement et ne nécessitent absolument aucune connaissance du Basic.

Si, de surcroît, vous faites partie de ceux qui souhaitent aborder la programmation des jeux sur Amstrad, ce livre vous en fournira toutes les clés. D'une part, les chapitres « annexes » vous initieront aux techniques de base spécifiques à ce genre de programmes (hasard, animation, lecture du clavier, examen du contenu de l'écran). D'autre part, les explications très détaillées qui accompagnent chaque jeu vous feront découvrir toutes les « ficelles du métier ».

Quoiqu'il en soit, donnez libre cours à votre imagination et ...

A vous de jouer !


EYROLLES

8375


EYROLLES
FAITES VOS JEUX avec
AMSTRAD
C.DELANNOY



Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE

https://acpc.me/